Introduction to Core Data

Session 123

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Introduction to Core Data

Matthew Firlik
Manager, Core Data

http://developer.apple.com/wwdc2005/

What Is Core Data?

http://developer.apple.com/wwdc2005/

What Is Core Data?

Sometimes, a mouthful…

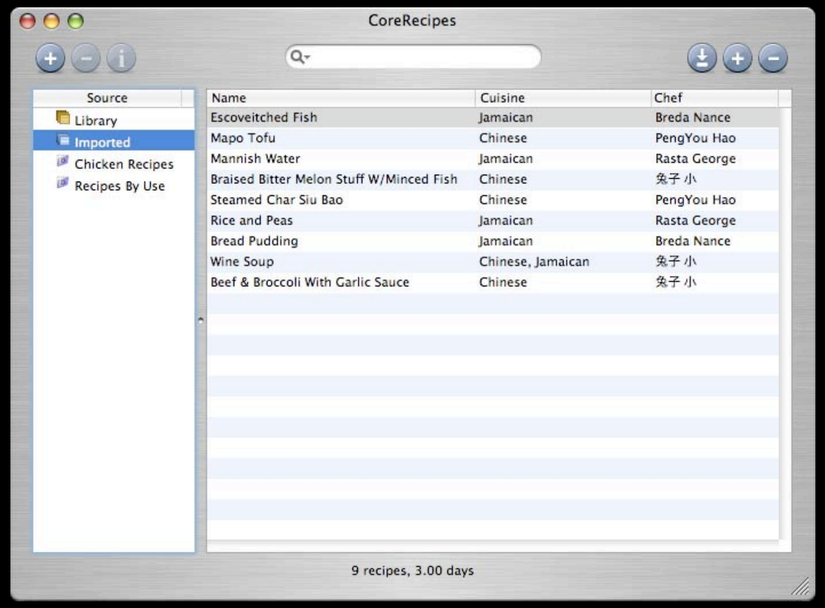
“Core Data completes the Cocoa vision for building well-factored applications based on the Model-View-Controller (MVC) pattern, and provides a strong, flexible, and powerful model-driven object graph management and persistency framework… “

- **Developing with Core Data**

  http://developer.apple.com/macosx/coredata

What Does Core Data Do?

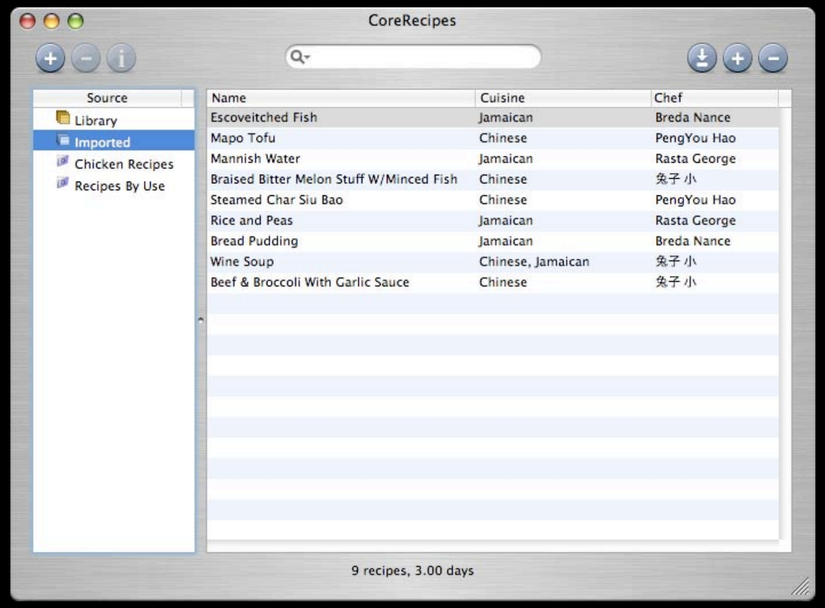
Provides a framework to interact with your data



- Visualization and display
- Creation, modification, and deletion
- Save and retrieval

What Does Core Data Do?

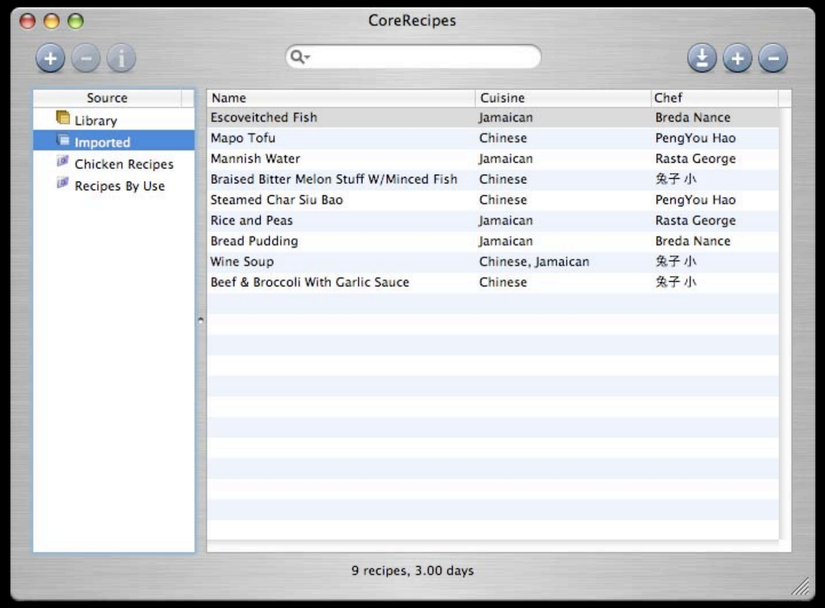
Provides a framework to interact with your data



- Visualization and display
- Object (graph) management
- Save and retrieval

What Does Core Data Do?

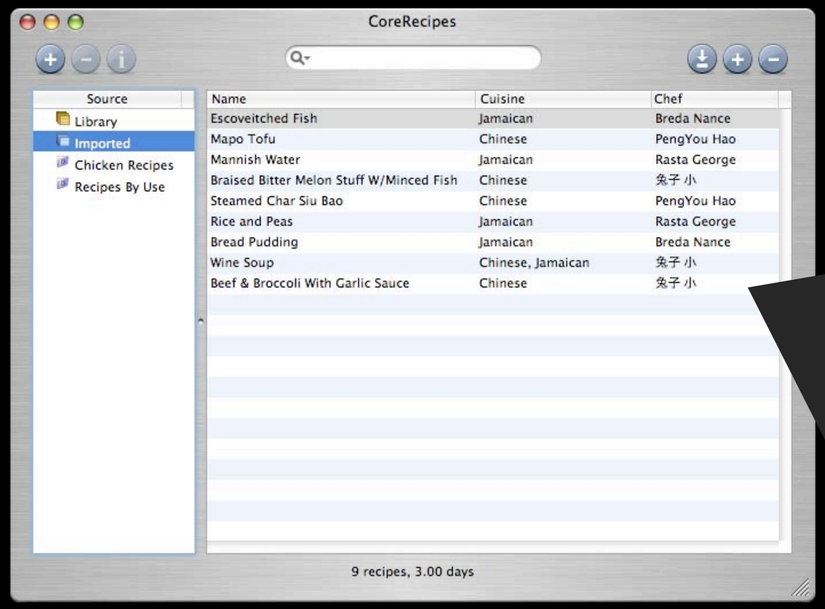
Provides a framework to interact with your data

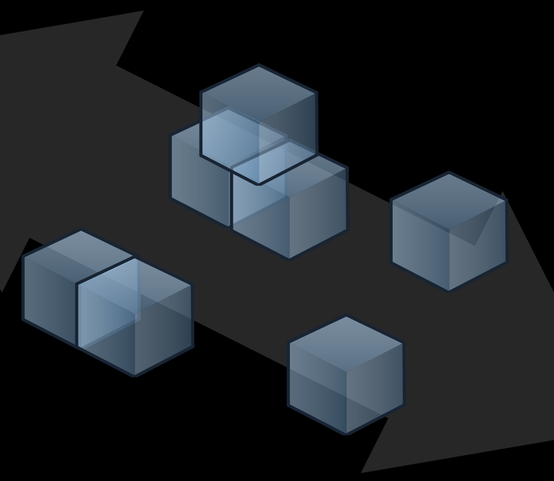


- Visualization and display
- Object (graph) management
- Persistence

What Does Core Data Do?

Down to the basics…



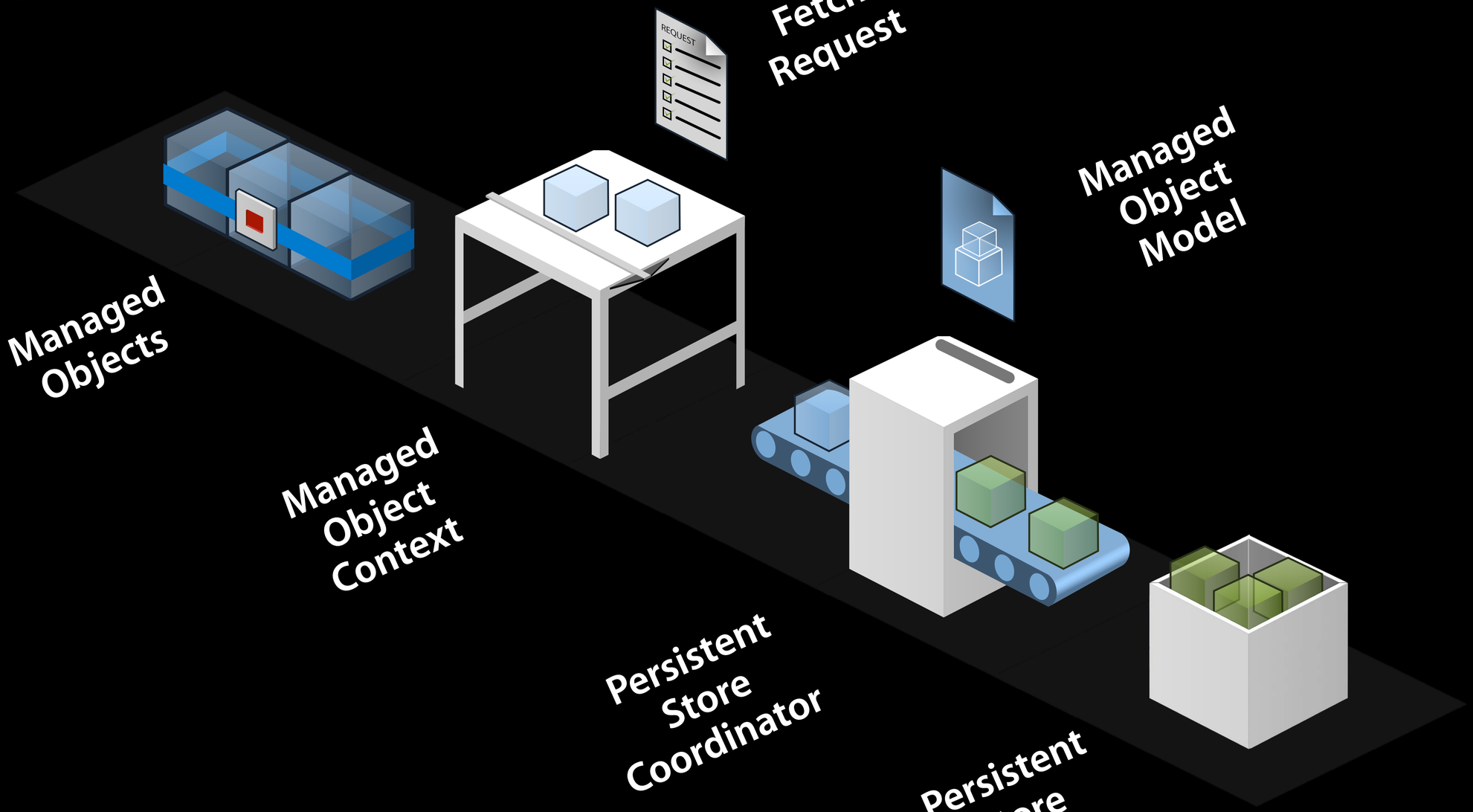


managed objects

```
<object type="RECIPE" id="z137">
     <attribute name="cookingtime"
 type="int16">10</attribute>
        <attribute
name="detaildescription"
type="string">A real treat for a
Sunday brunch.  We usually serve
this with a light sprinkle of
powdered sugar and warm maple
syrup.</attribute>
        <attribute name="name"
type="string">French Toast</
```

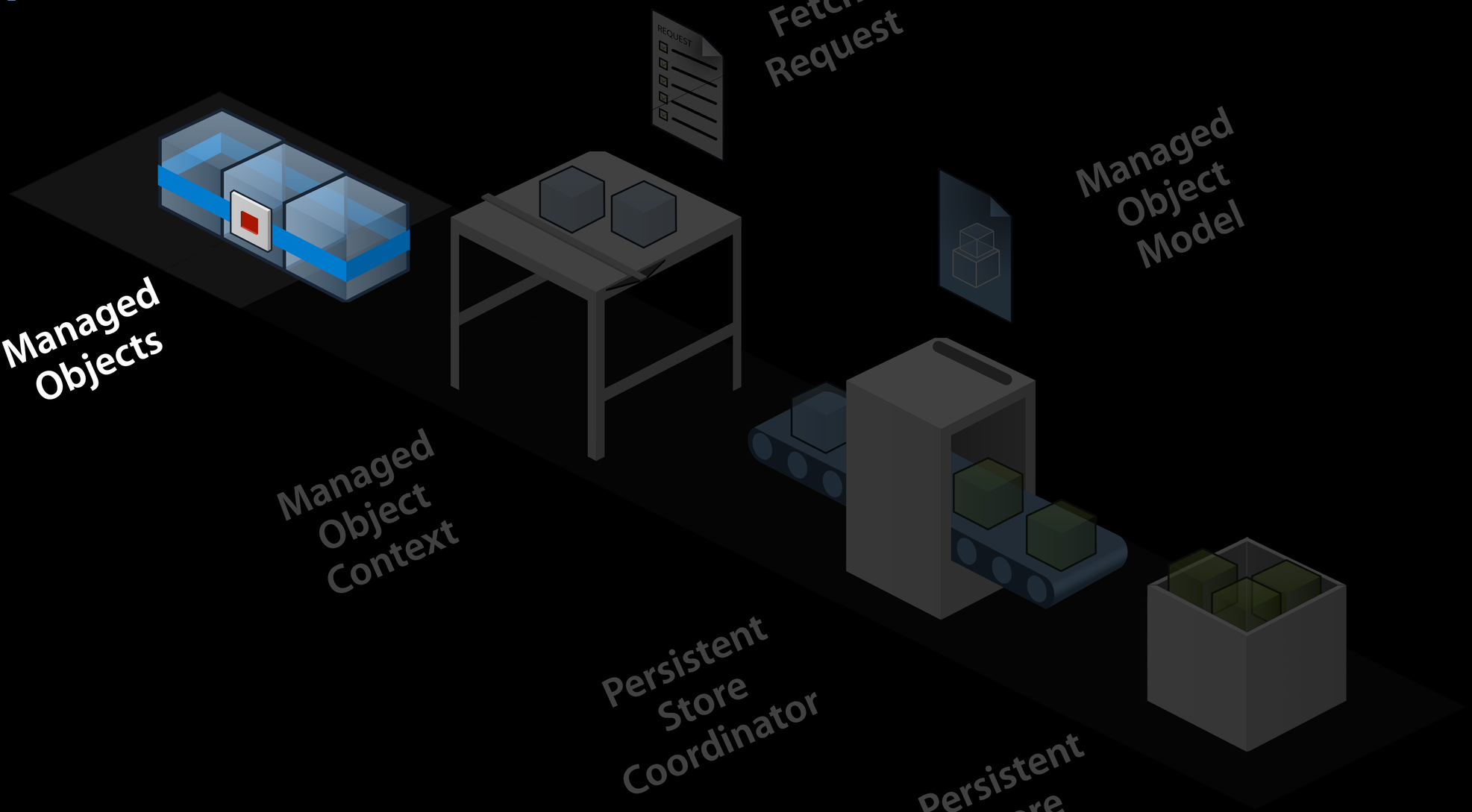
Core Data: The Architecture

The "persistence stack"



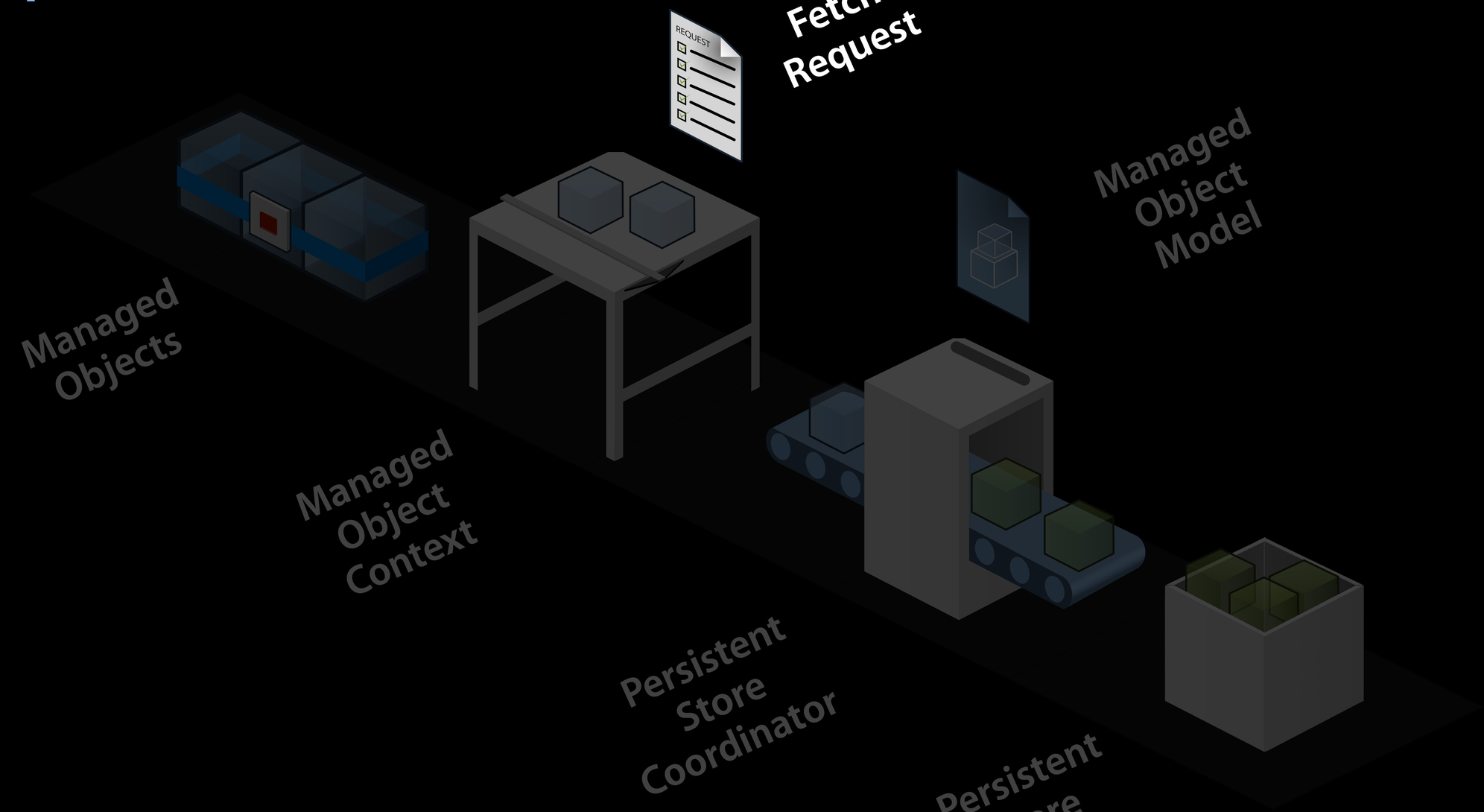
Core Data: The Architecture

The "persistence stack"



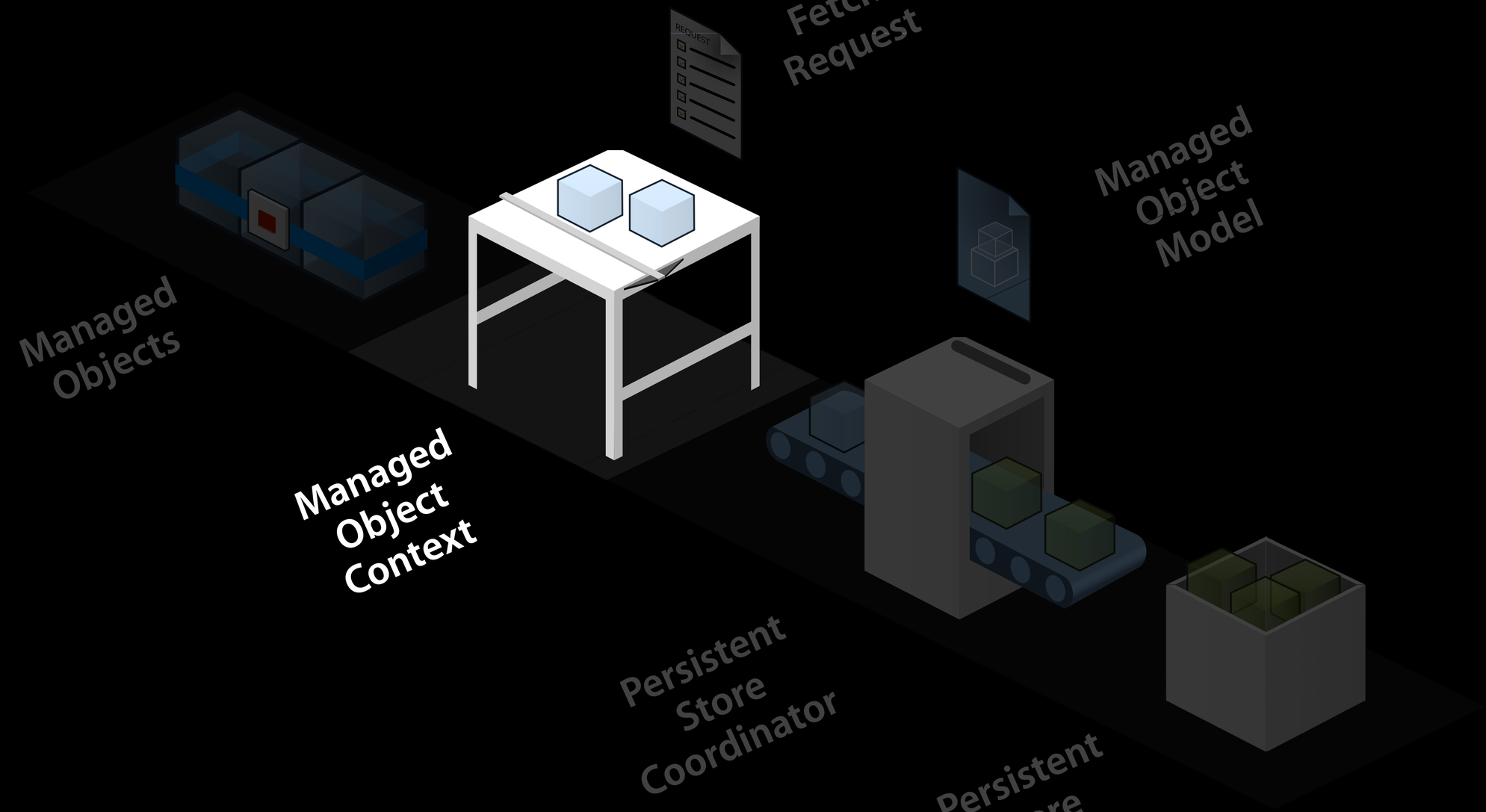
Core Data: The Architecture

The "persistence stack"



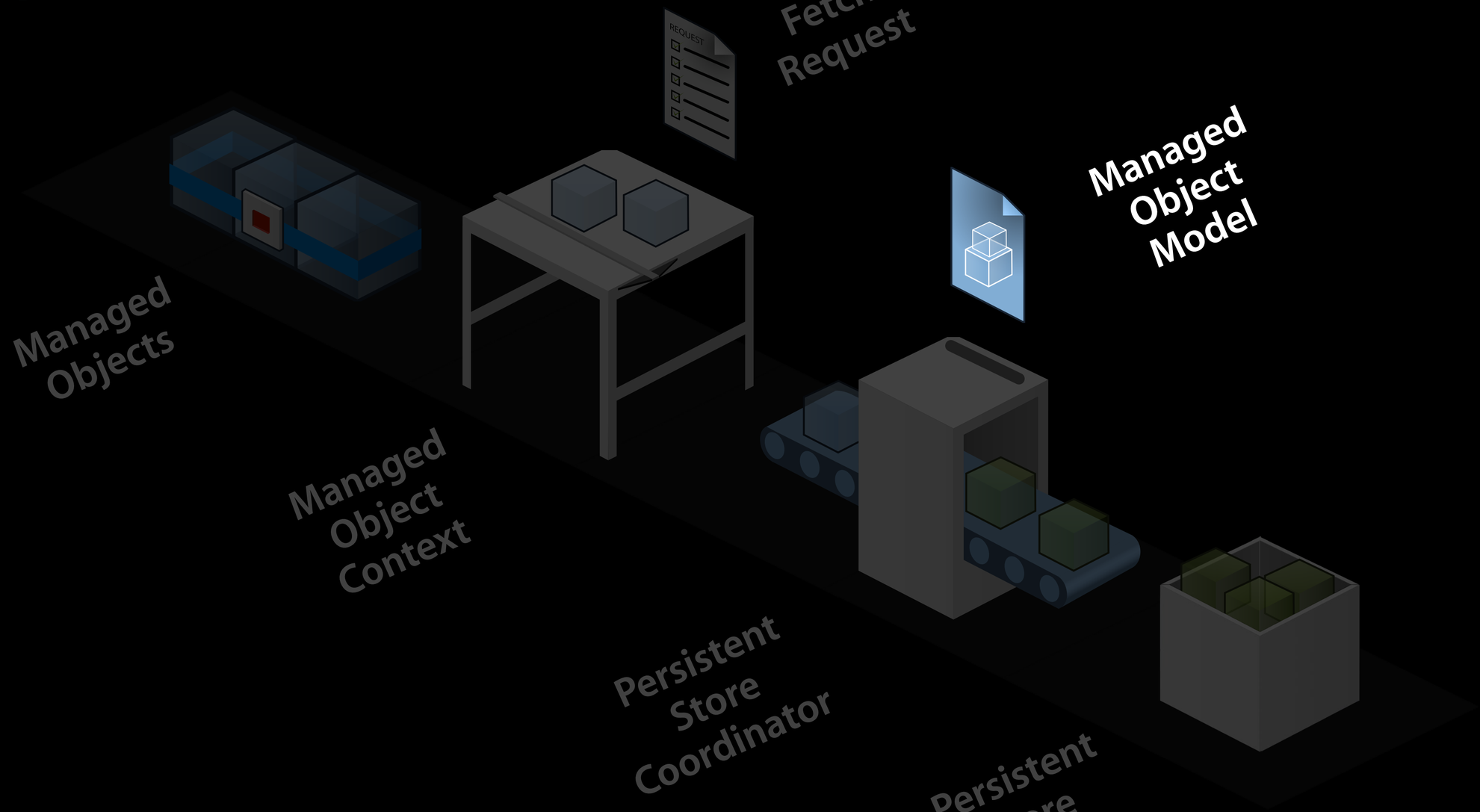
Core Data: The Architecture

The "persistence stack"



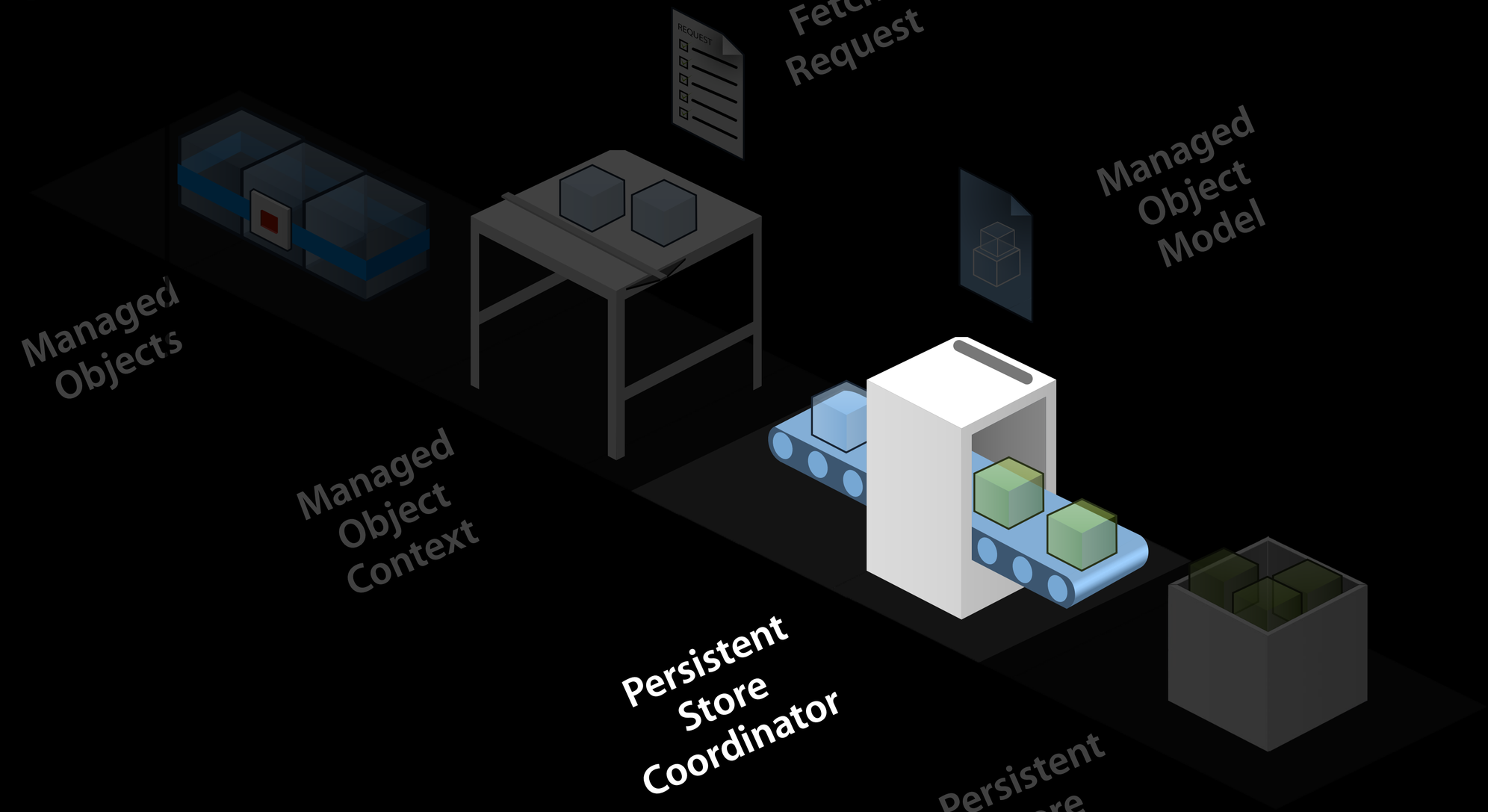
Core Data: The Architecture

The "persistence stack"



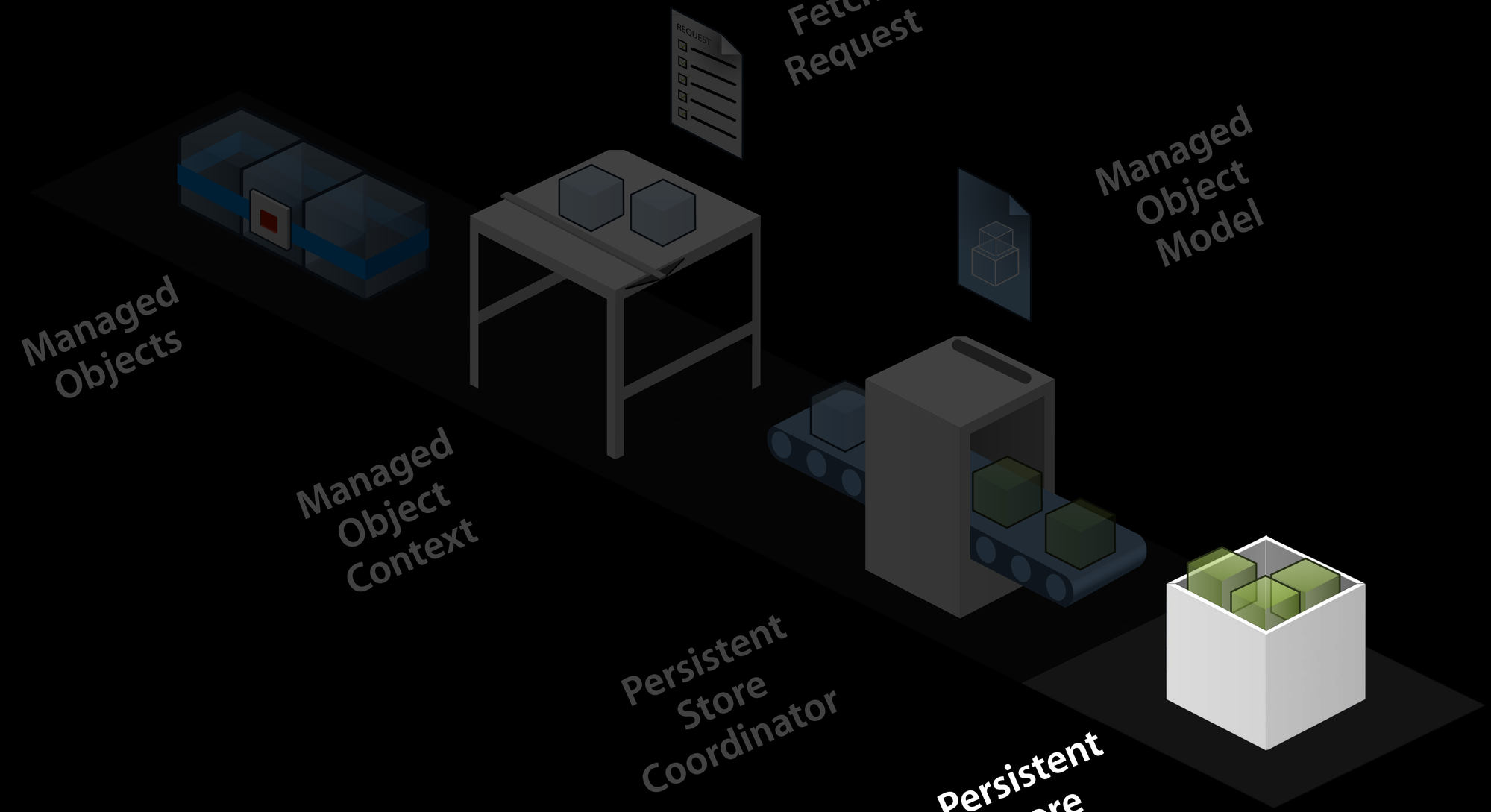
Core Data: The Architecture

The "persistence stack"



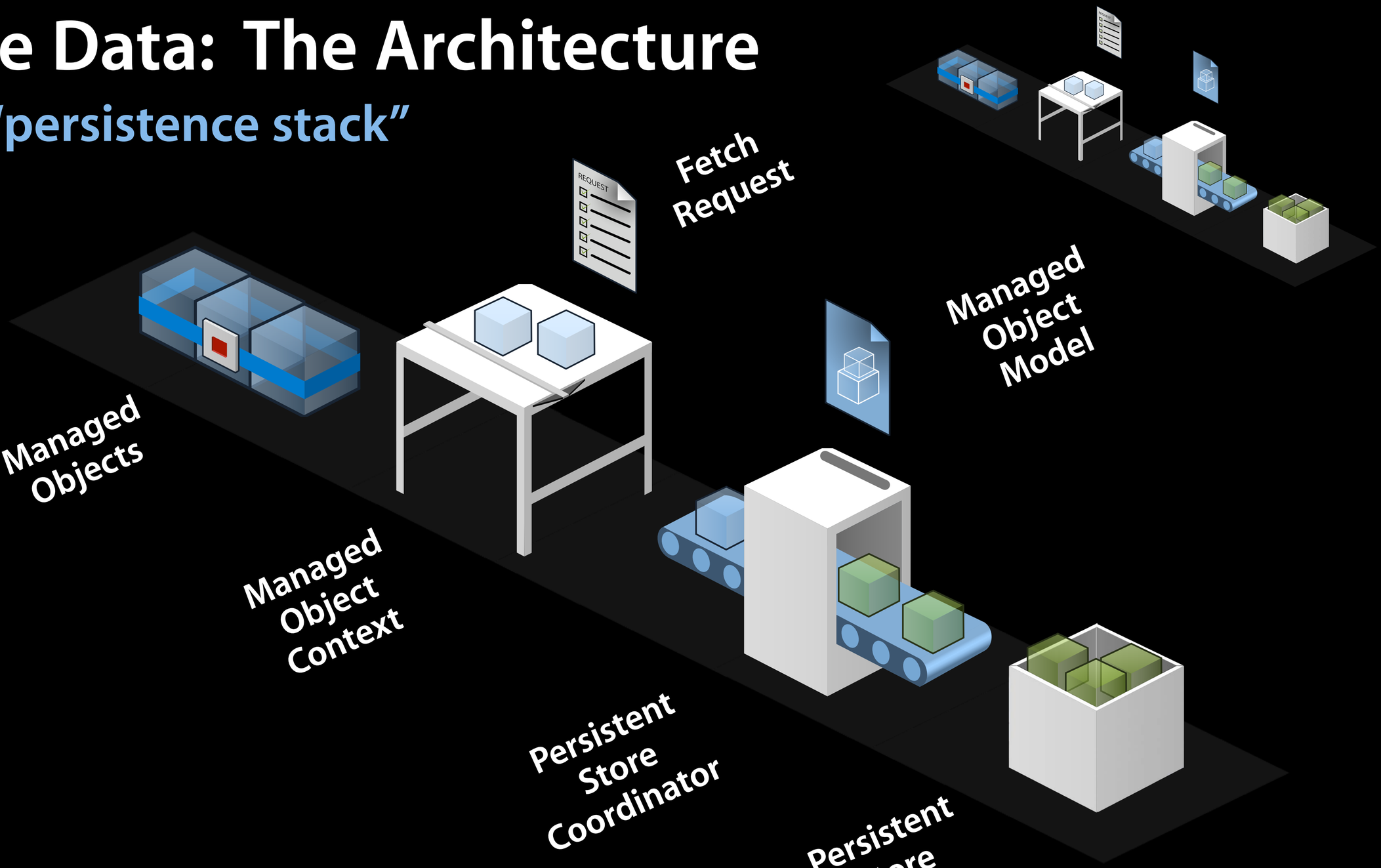
Core Data: The Architecture

The "persistence stack"



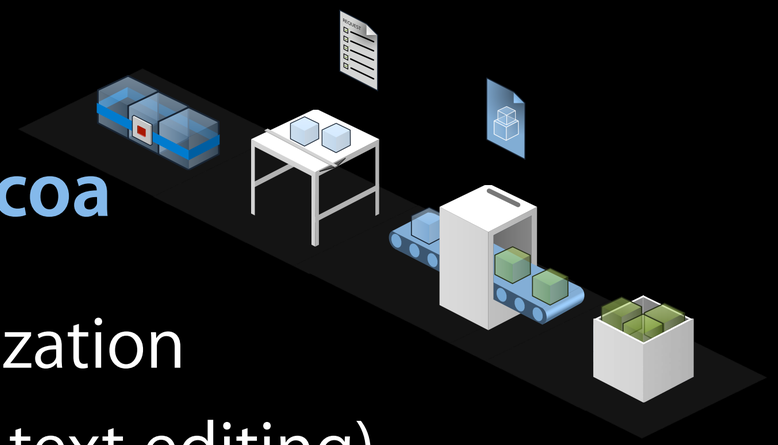
Core Data: The Architecture

The "persistence stack"



What Does Core Data Do?

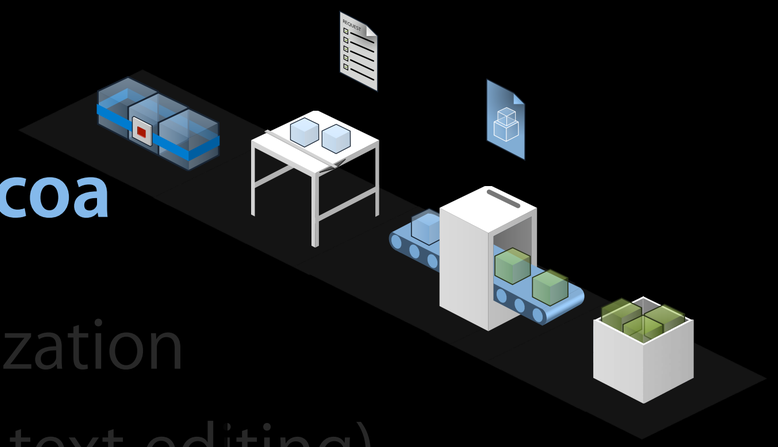
Object management and persistency for Cocoa



- Integration with Cocoa bindings for UI synchronization
- Built-in management of undo and redo (beyond text editing)
- Automatic validation of property values
- Grouping, filtering, and organizing data
- Change propagation, including maintaining consistency of relationships
- Automatic support for storing objects in external data repositories
- Several store options (SQL, XML, binary)

What Does Core Data Do?

Object management and persistency for Cocoa

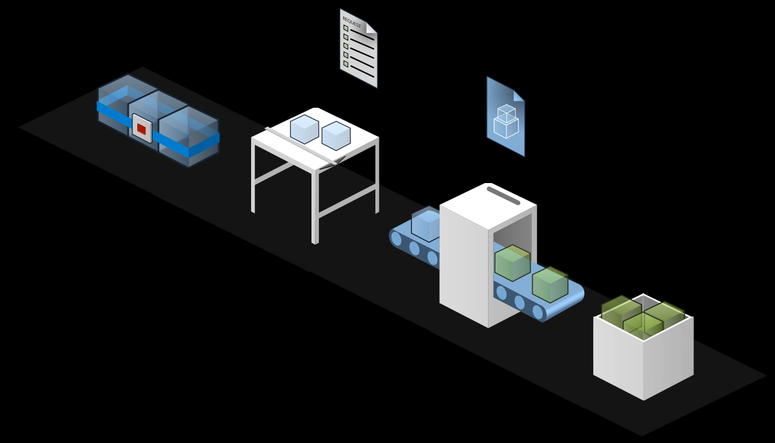


- Integration with Cocoa bindings for UI synchronization
- Built-in management of undo and redo (beyond text editing)
- Automatic validation of property values
- Grouping, filtering, and organizing data
- Change propagation, including maintaining consistency of relationships
- Automatic support for storing objects in external data repositories
- Several store options (SQL, XML, binary)

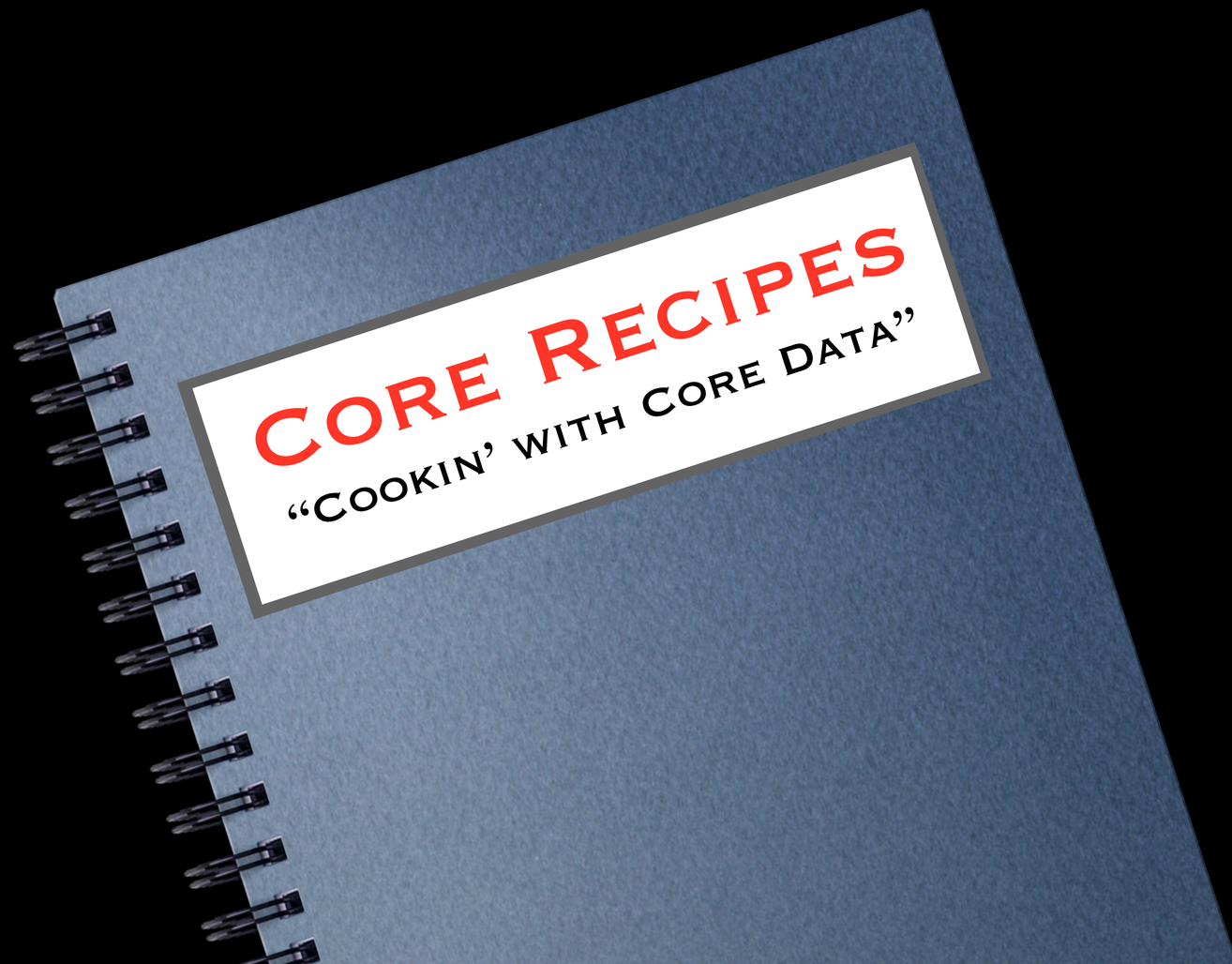
Use a much as you need

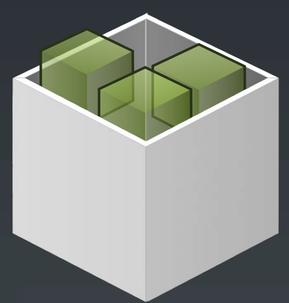
WWDC Sample Code

“CoreRecipes”



- Hands-on samples (for tomorrow’s session)
  - Command-line and GUI applications
- CoreRecipes application
  - Core Data essentials
  - Cocoa bindings
  - Spotlight integration
  - Automator action
- Data migration sample





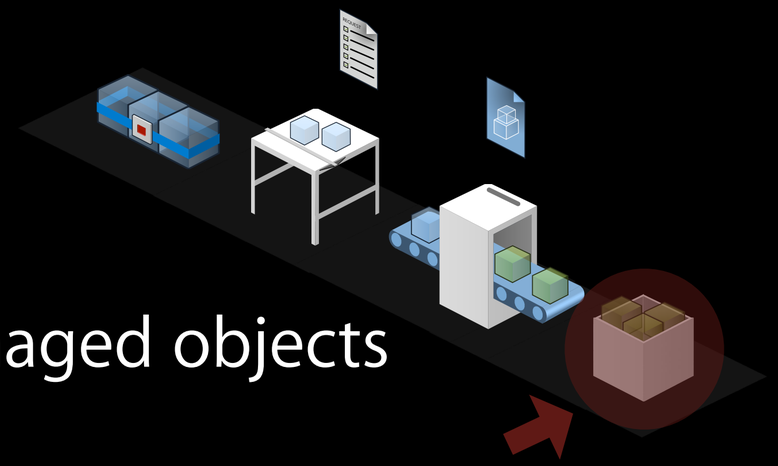
The Persistent Store

Where your data comes and goes

http://developer.apple.com/wwdc2005/

The Persistent Store

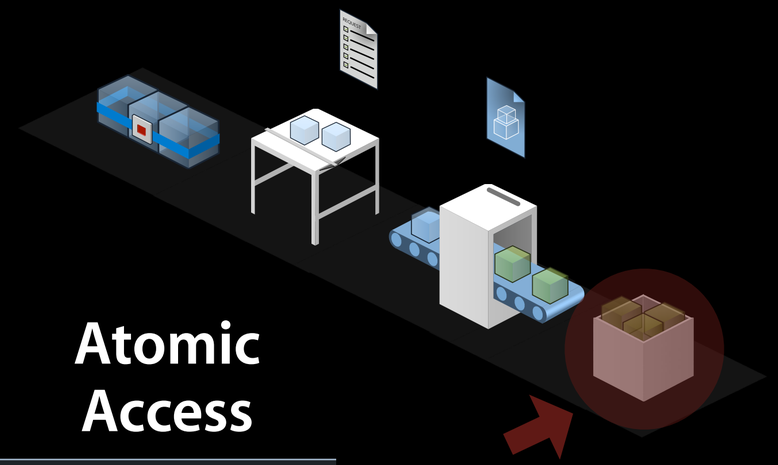
Where your data comes and goes



- Logically, a file or collection of files in which managed objects may be stored
- Transparent to the application and developer
  - Each store has a Universally Unique Identifier (UUID) and type
  - Currently, no API to work with stores directly
  - Use the persistent store coordinator to interact

Persistent Storage Options

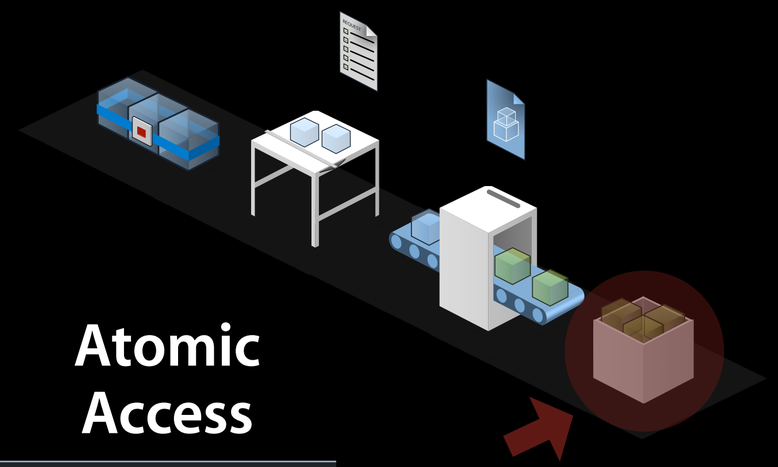
Types and benefits



	Scalability	Speed	Human Readable	Atomic Access
XML		Good	✓	✓
Binary		Better		✓
SQLite	✓	Best		

The In-Memory Store

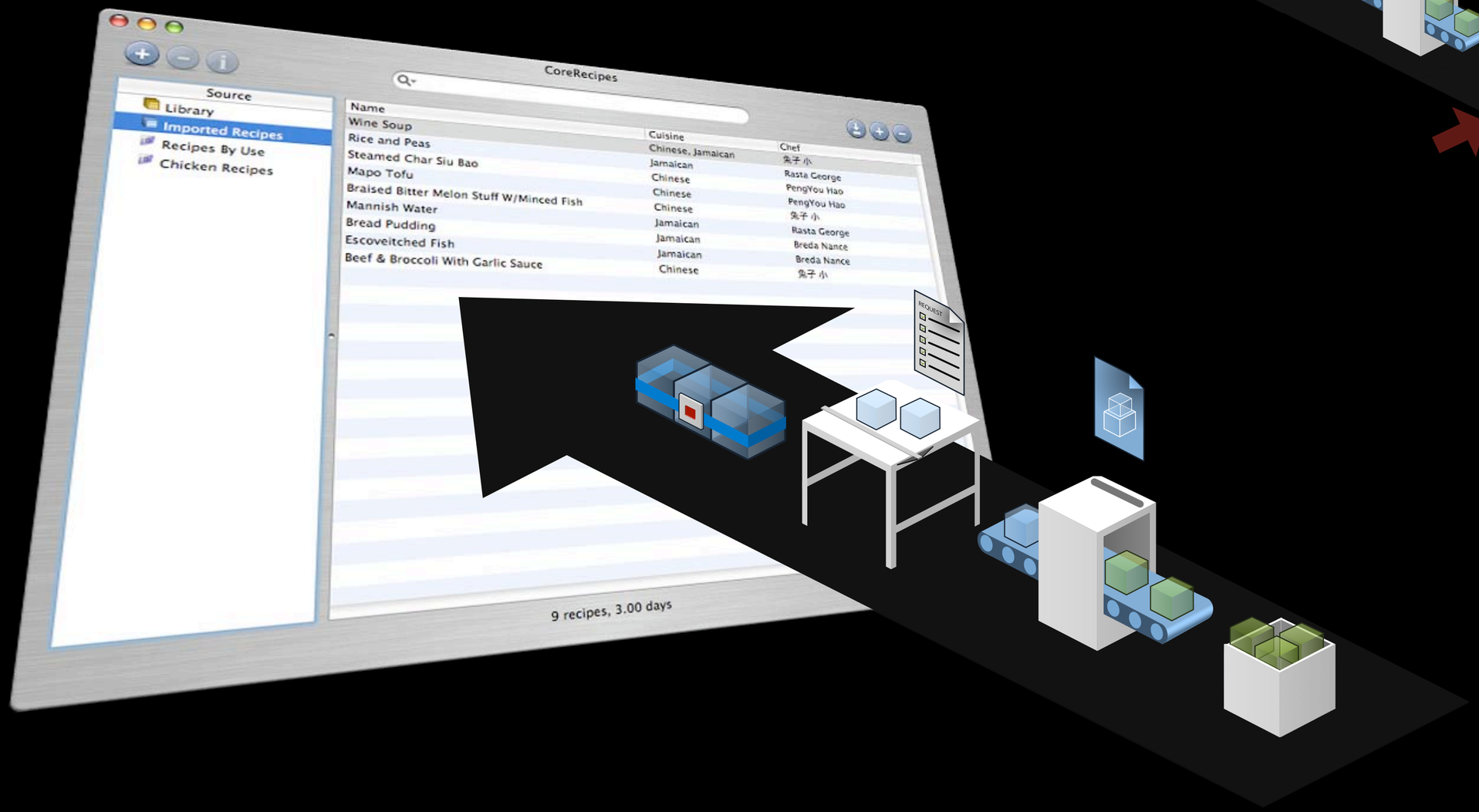
Use with existing files or transient data



	Scalability	Speed	Human Readable	Atomic Access
XML		Good	✓	✓
Binary		Better		✓
SQLite	✓	Best		
In-Memory				

How Many Stores to Use?

Some applications use one…



How Many Stores to Use?

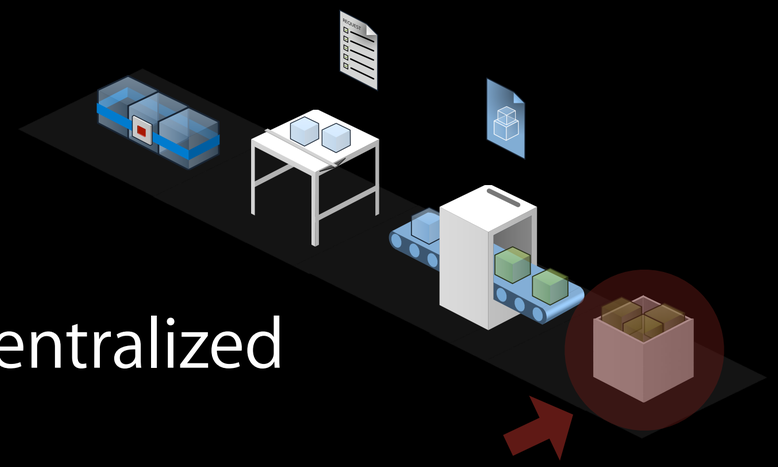
Some applications use many, *collaborative*…

How Many Stores to Use?

Some applications use *separate* documents…

How Many Stores Should I Use?

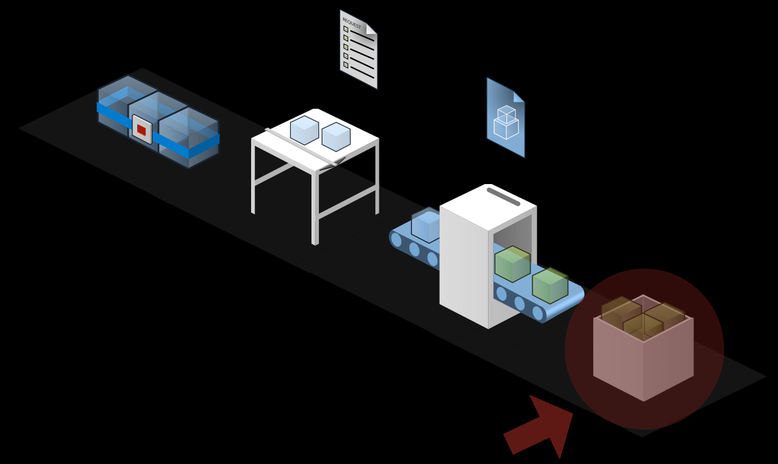
It depends on what you are designing…



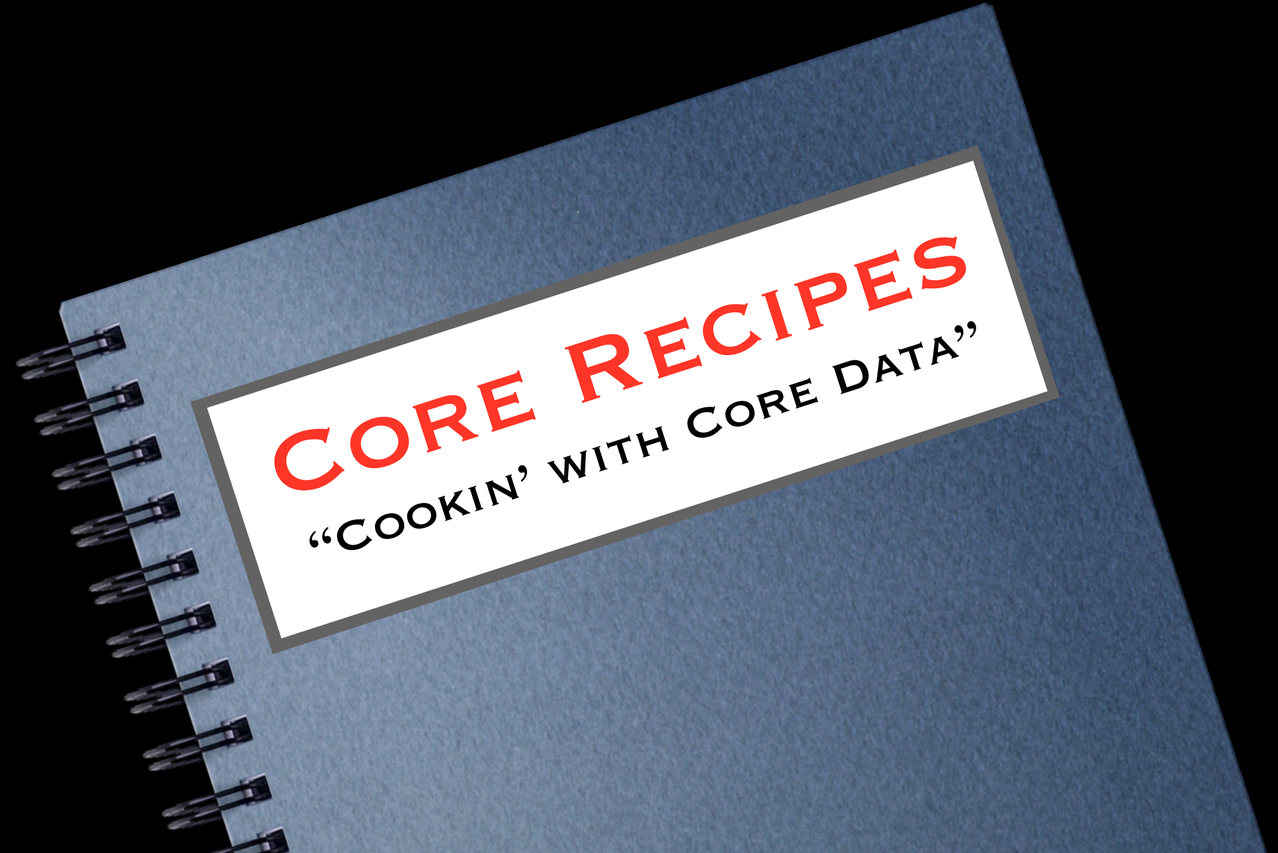
- Single stores for applications where the data is centralized and the information is inter-related
- Multiple stores for projects that might need to "share" data
  - Be careful about cross-store dependencies
- Document-based stores for data that is completely encapsulated

Core Recipes App

A tough decision



- Recipes as separate documents?
- All recipes in a single store?
- Implemented as a multiple store application



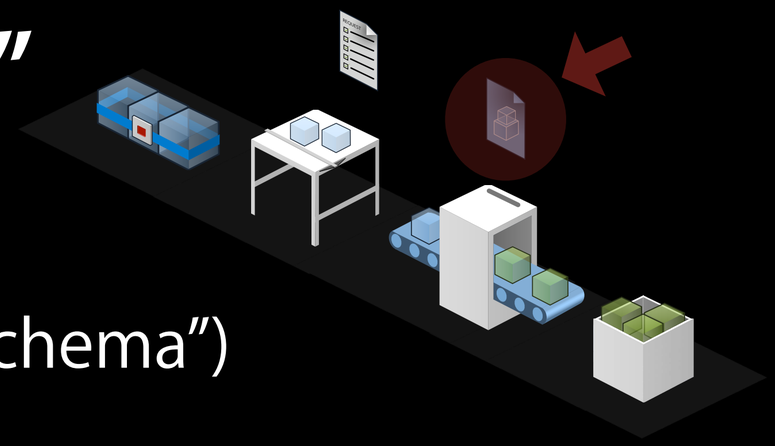
The Managed Object Model

Your data blueprint

http://developer.apple.com/wwdc2005/

What Do You Mean, "Modeling?"

Creating a blueprint for your data

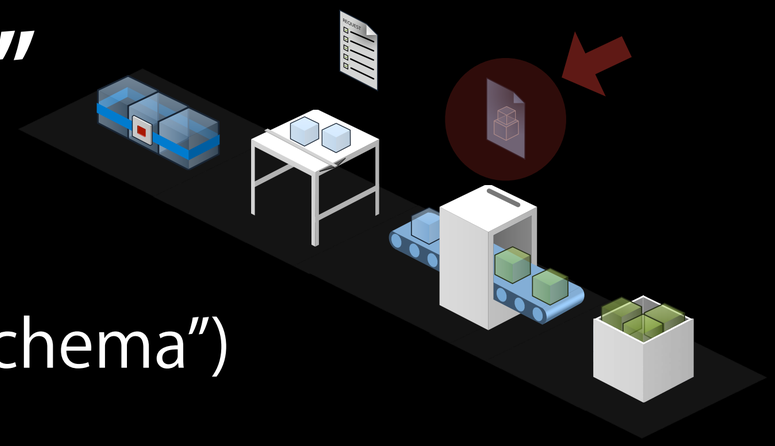


- A managed object model is a high-level view ("schema") of the data for your application

Entity	Classification of Object Runtime representation
Property	Characteristics of an entity Attributes or relationships

What Do You Mean, "Modeling?"

Creating a blueprint for your data

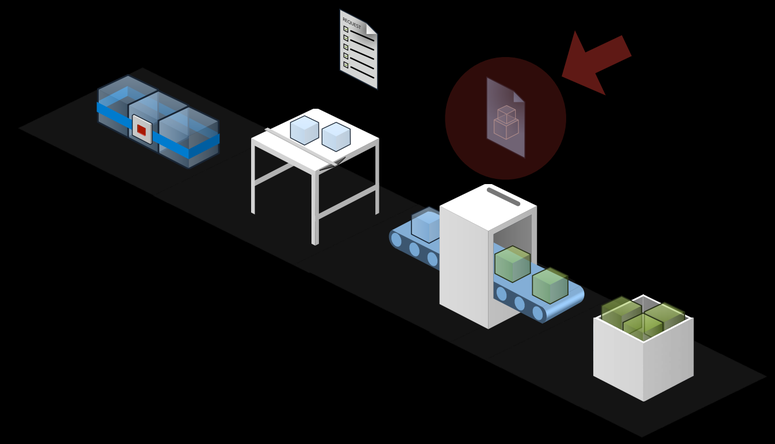


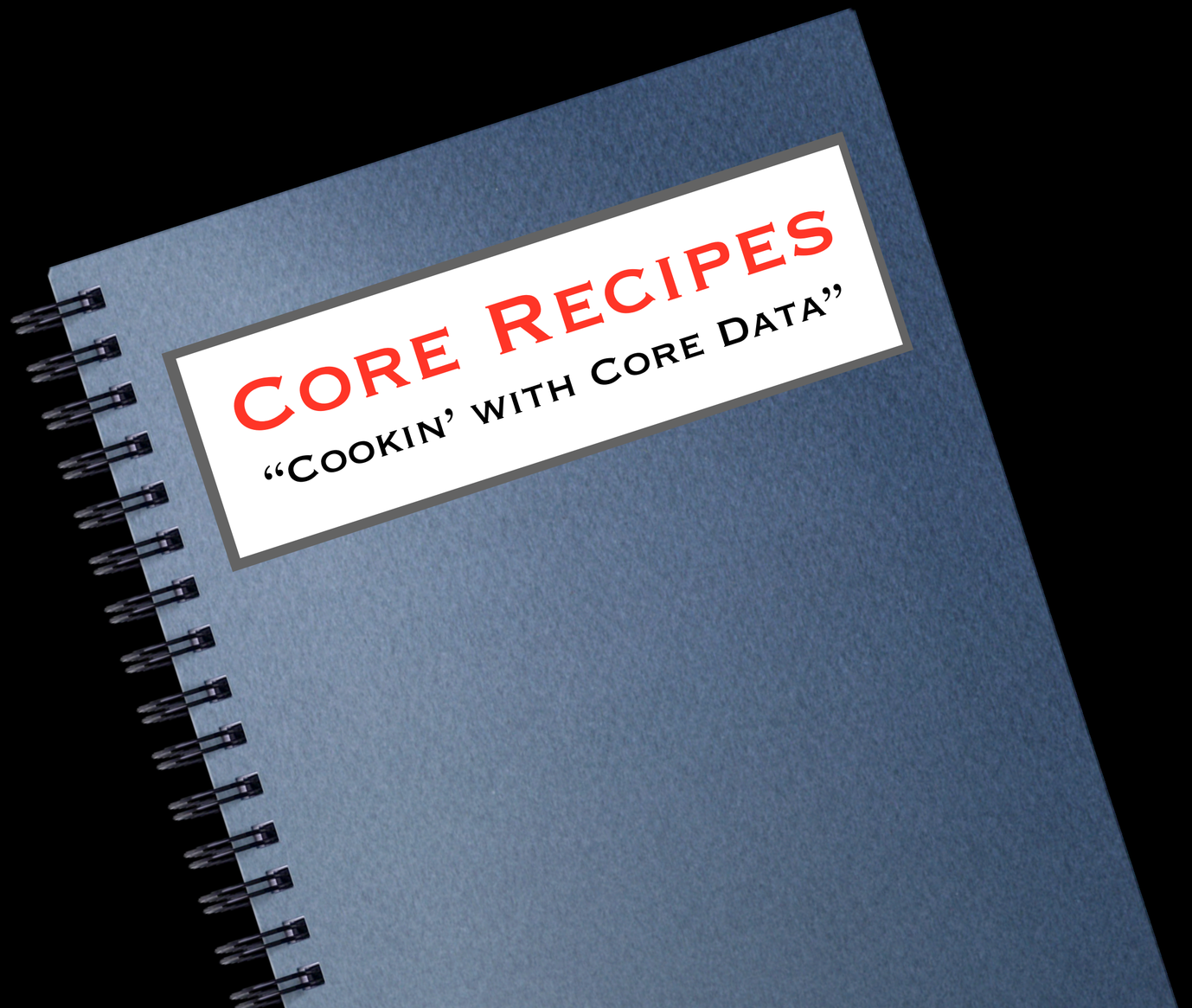
- A managed object model is a high-level view ("schema") of the data for your application

Entity	Classification of Object Runtime representation
Attribute	Simple instance variable
Relationship	Managed references to other entities Singular ("to-one") or multiple ("to-many")

Core Recipes App

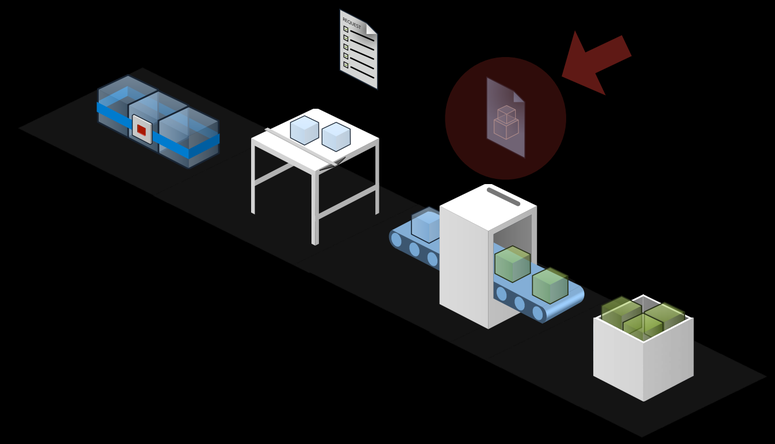
An example of decomposing data…

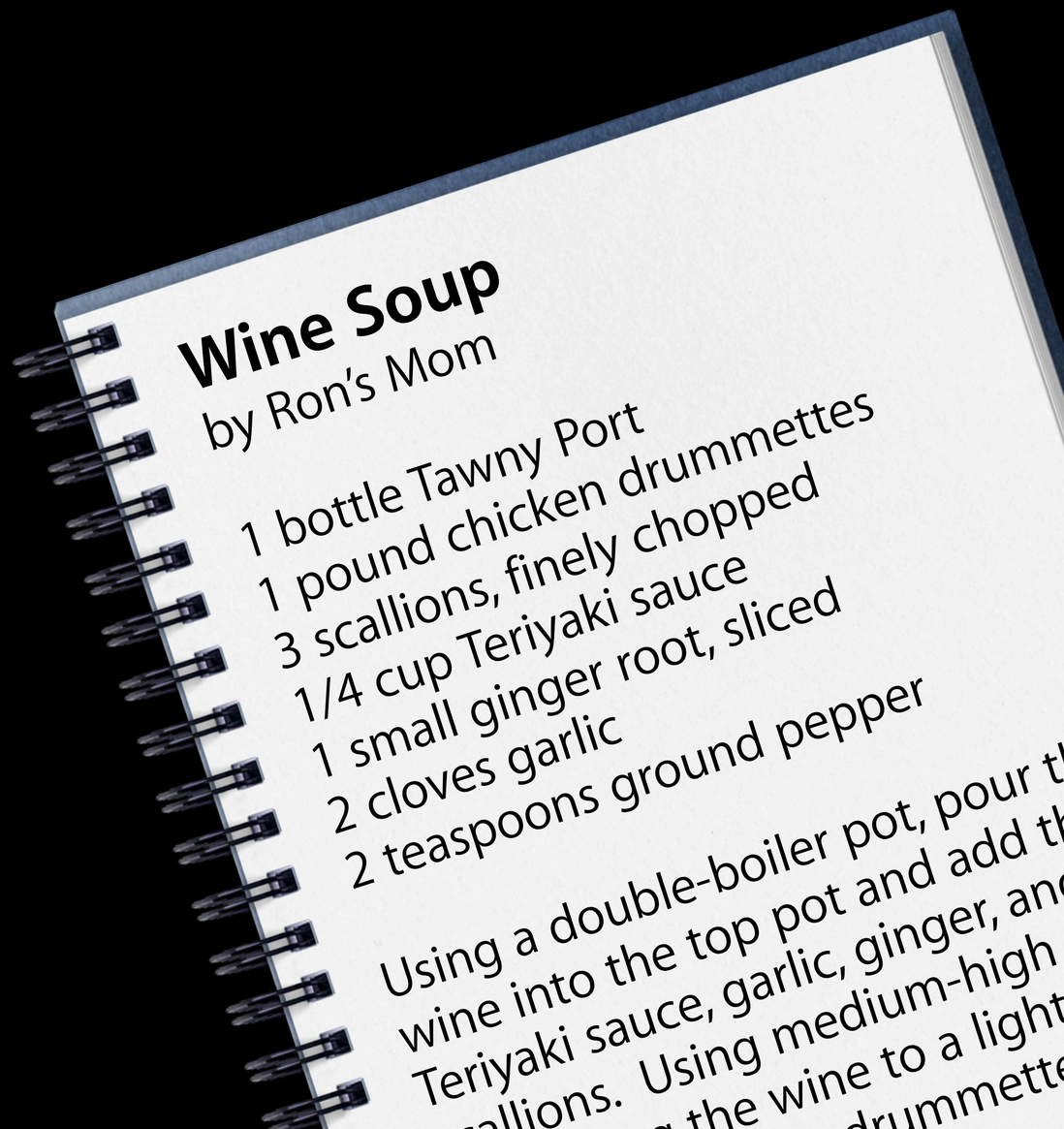




Core Recipes App

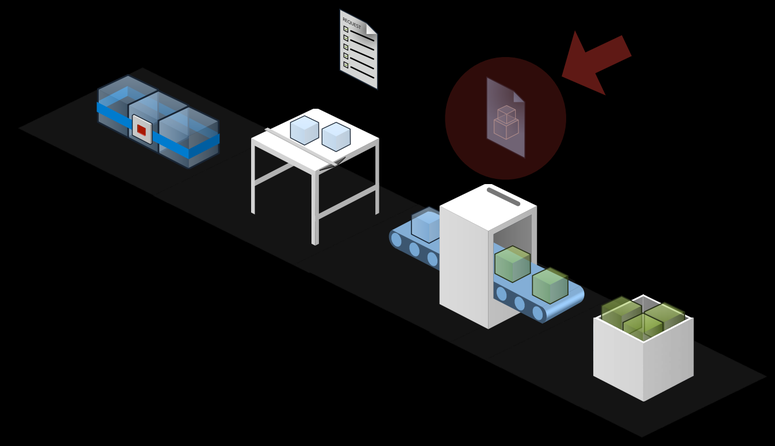
An example of decomposing data…

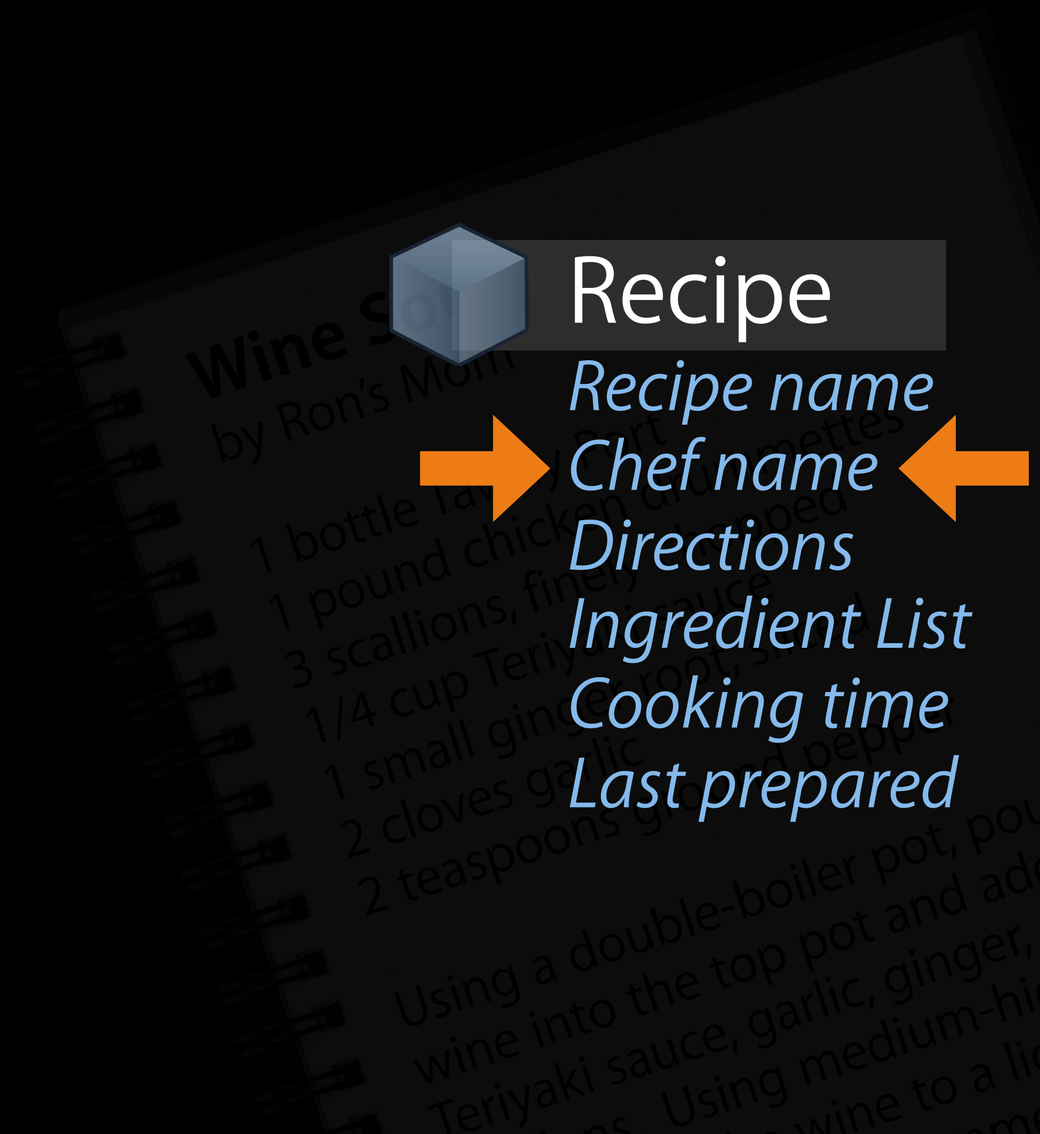




Core Recipes App

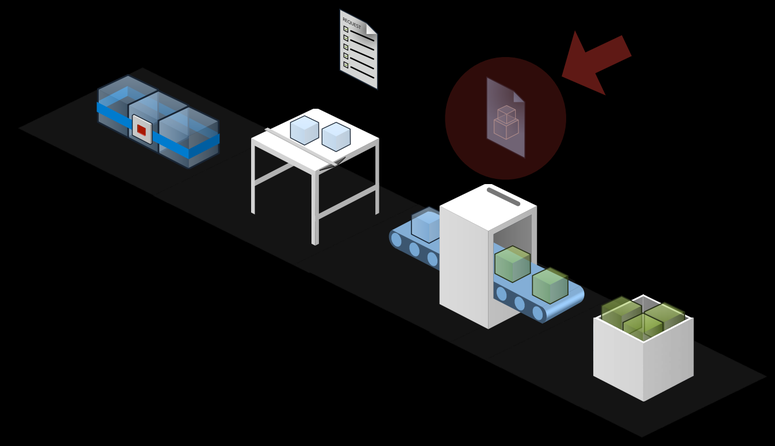
An example of decomposing data…

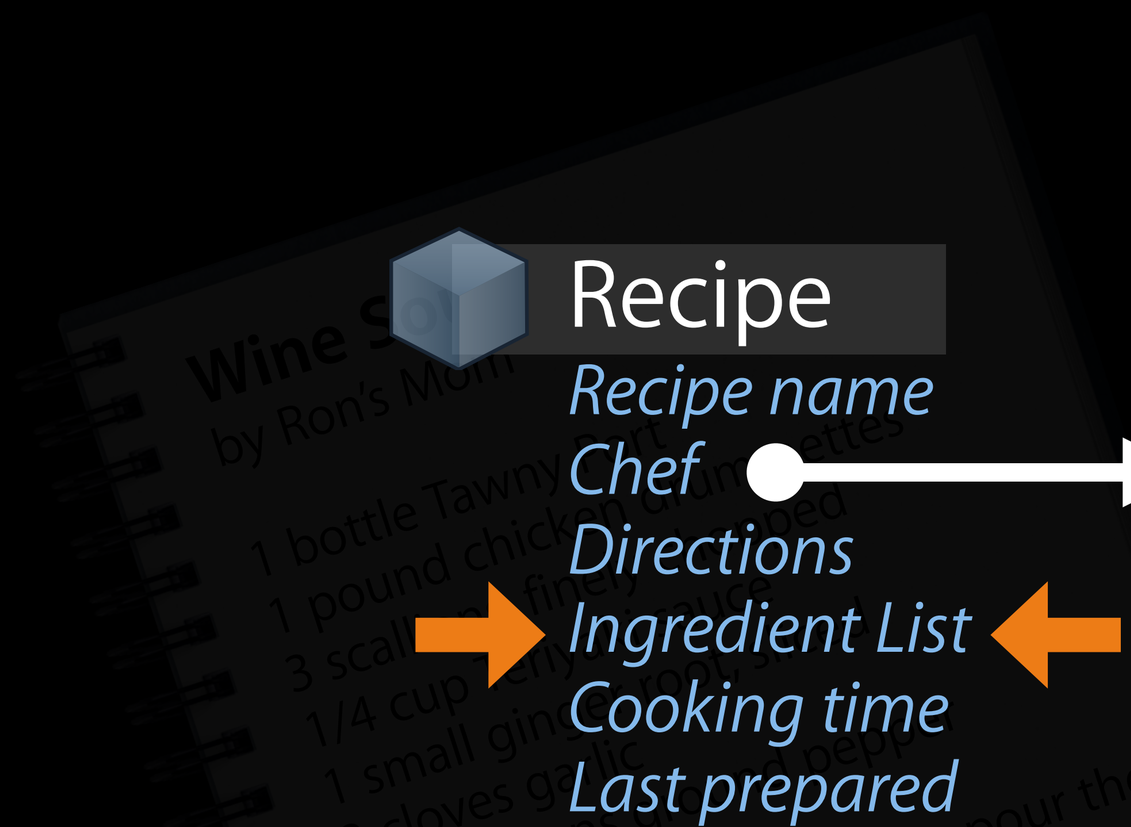




Core Recipes App

An example of decomposing data…



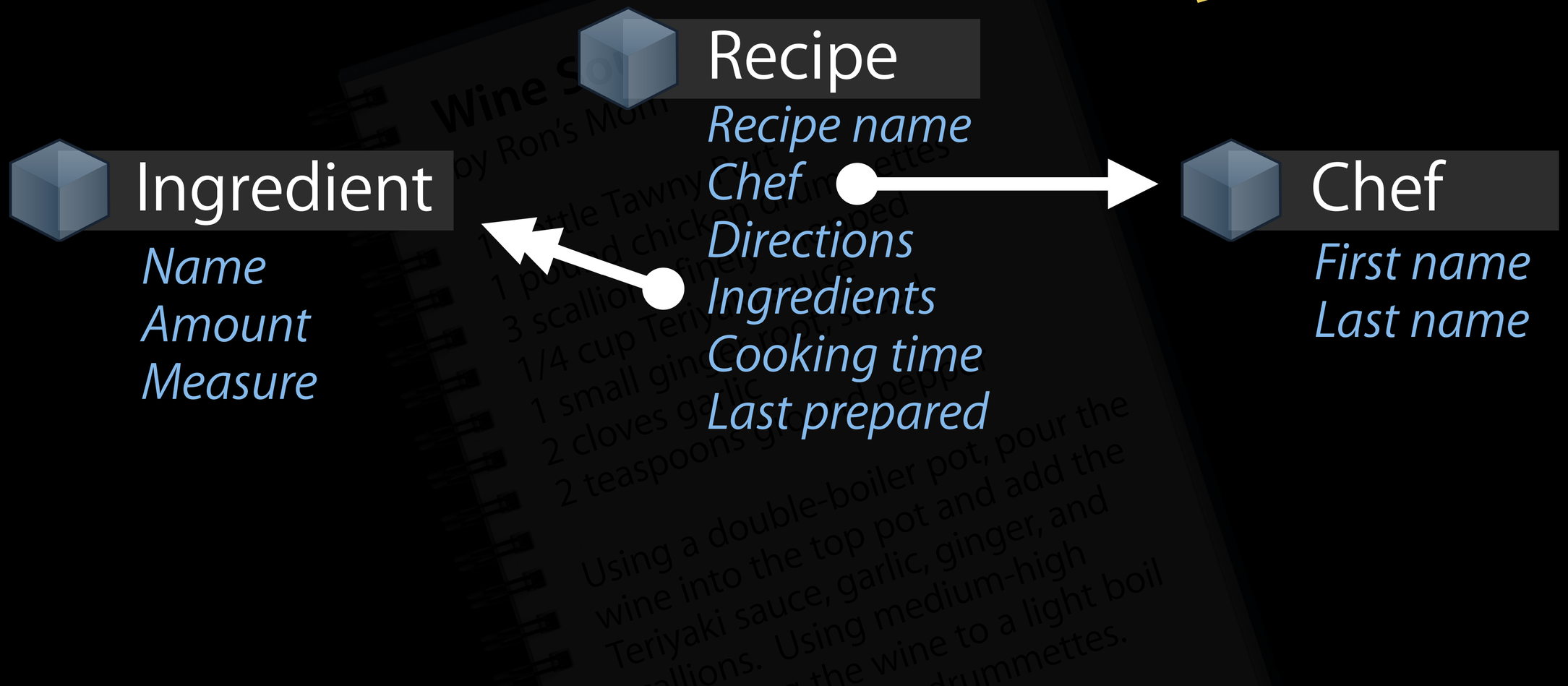


Core Recipes App

An example of decomposing data…

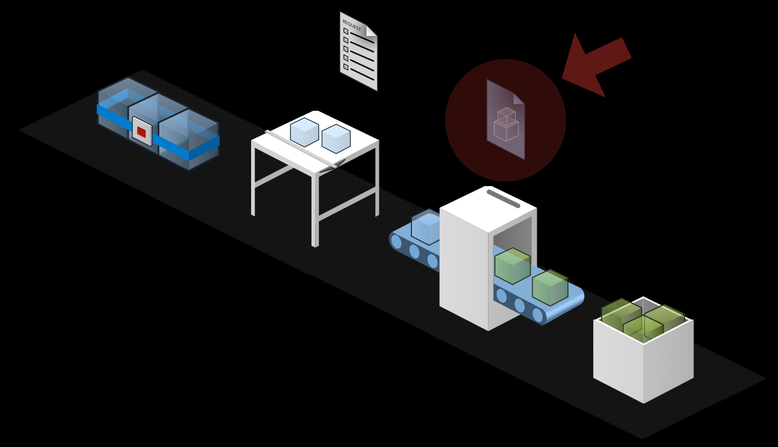
API Reminder

```
to-many relationships
always return sets, not
arrays
```



What Happens at Runtime?

The model is used as a blueprint…



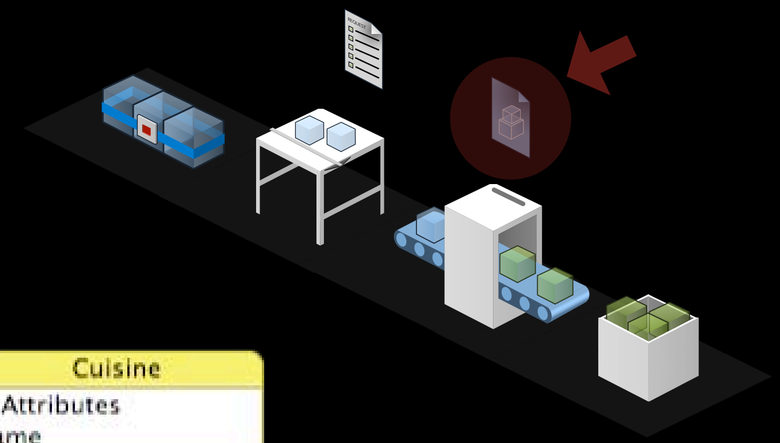
Recipe

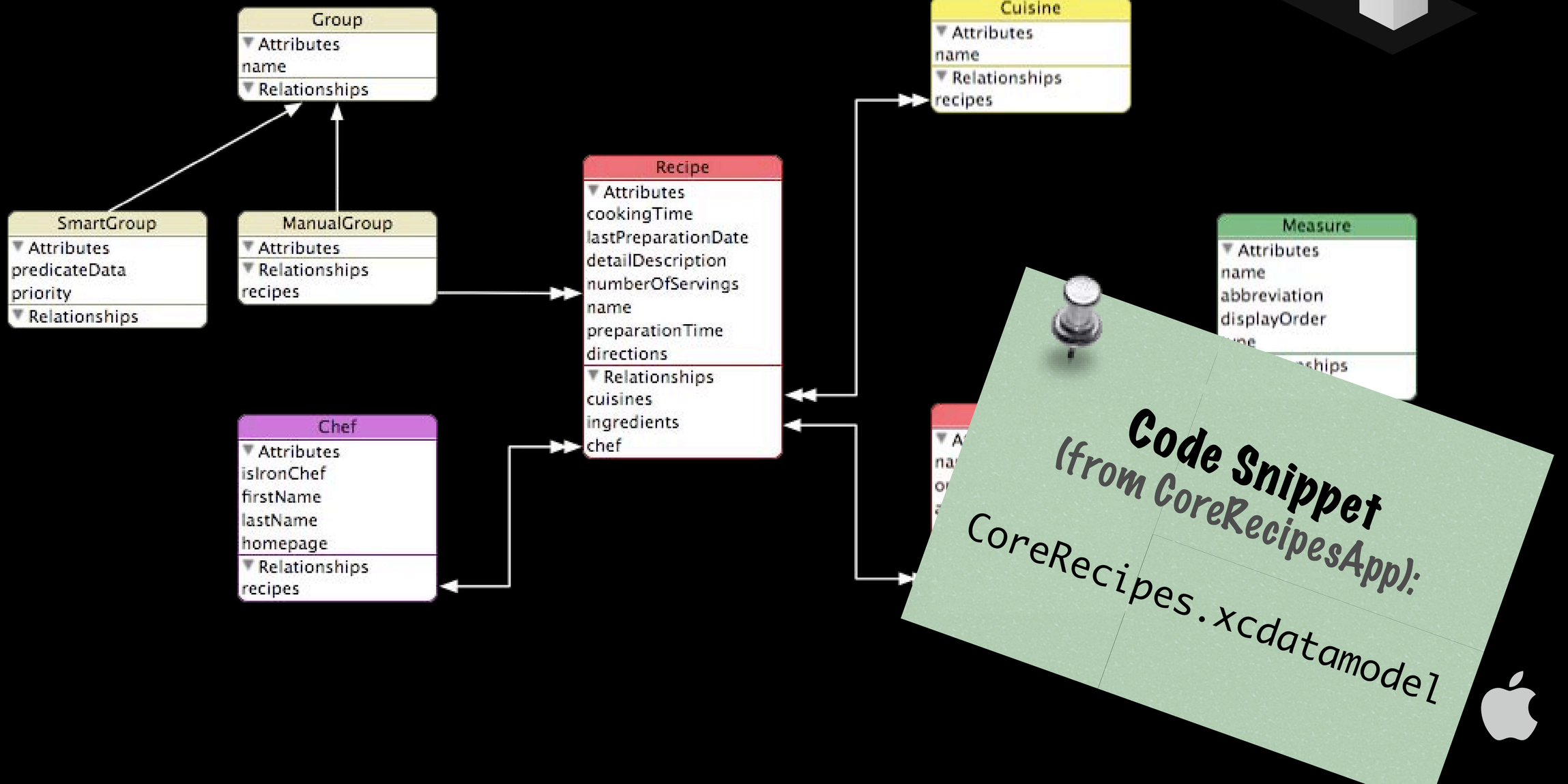
Recipe name
Chef
Directions
Ingredients
Cooking time
Last prepared

```
<Recipe: 0x451d8c0>
 (entity: Recipe; id: 0x38c130
<x-coredata://CBBE4927-854E-4277-88B0-A4A799582B
    recipeName = "Wine Soup";
    chef = 0x38b170 <x-coredata://CBBE4927-854E-4277
    directions = "Using a double-boiler pot, pour the wi
    ingredients = "<relationship fault: 0x451e000 'ingred
    cookingTime = 0;
    lastPrepared = 2005-05-26 02:07:16 -0700;
})
```

Core Recipes App

In the end, it might look like…





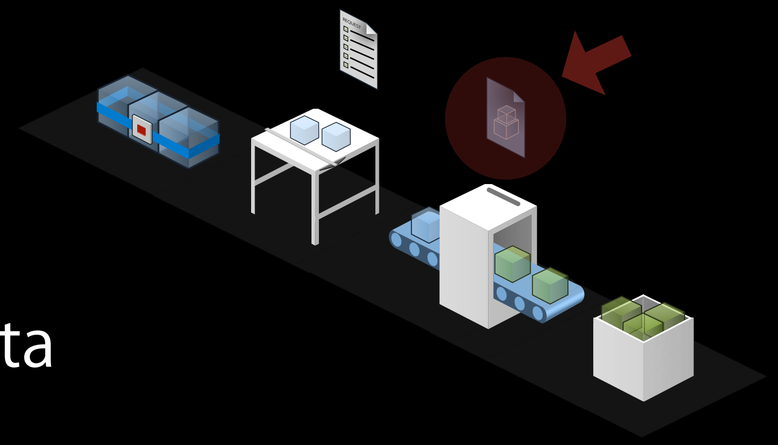
What Else Can My Model Do?

Quite a bit, actually…

- Define default attribute values
- Set-up validation rules
- Set-up deletion rules (for relationships)
- Create query templates
- Customization (using userInfo dictionaries)

How Important Is the Model?

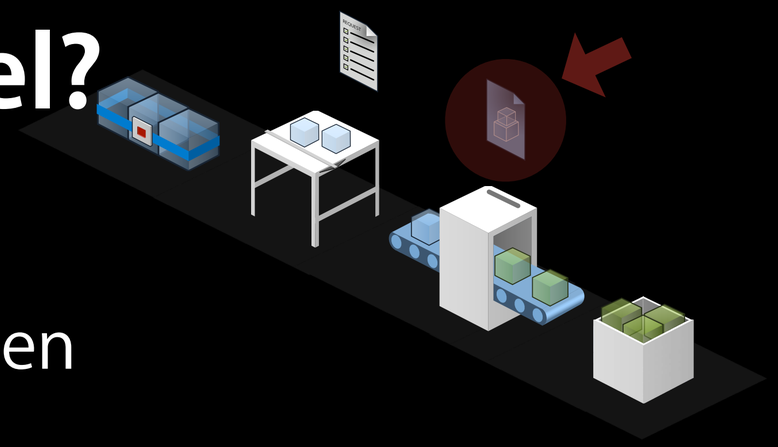
Absolutely critical



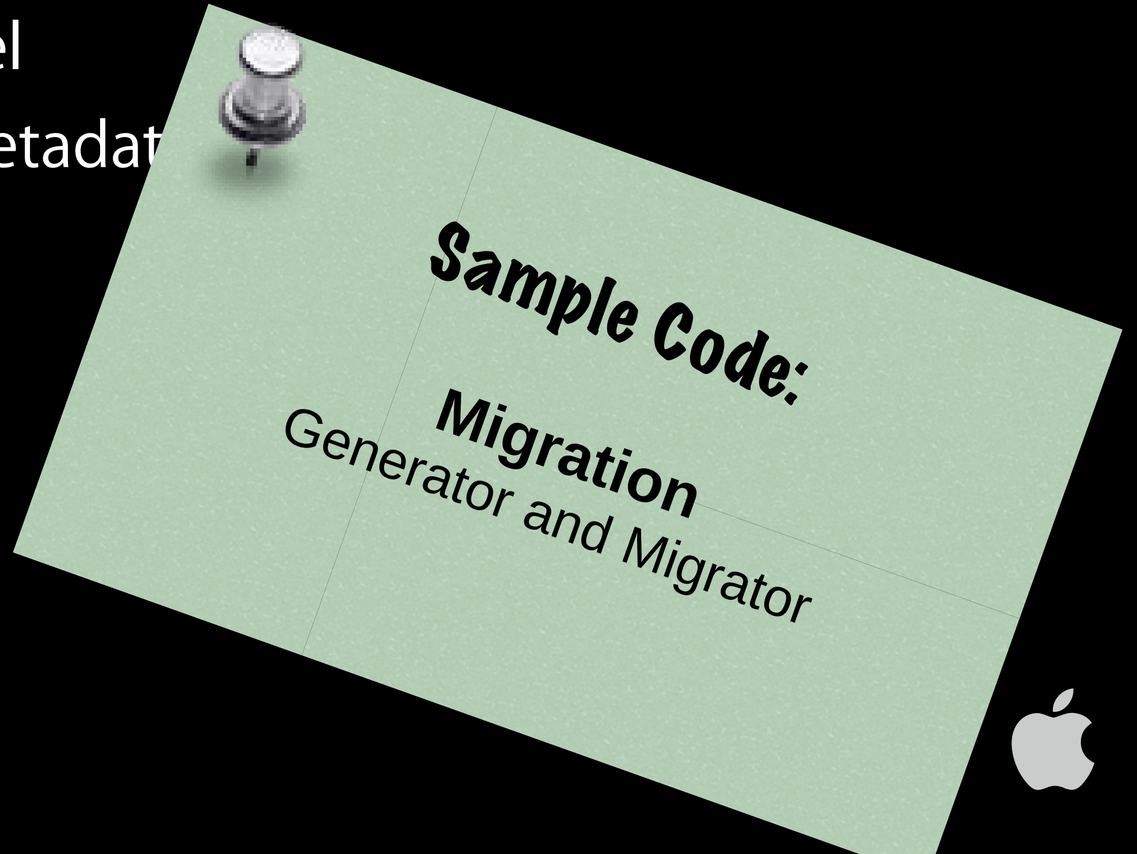
- Models are "contracts" between you and Core Data
  - Defines guidelines for data interaction
  - Outlines policies to maintain integrity
- Design with function **and** performance in mind
  - Consider both fine and coarse-grained objects
- The more complicated the model, the greater the value-add of Core Data

What About Changing My Model?

The question of versioning



- Making changes to your model after data has been saved in a persistent store is not supported
  - This is a general issue with “versioning”
- Some things to think about
  - Storing a version number with the model
  - Saving the model version in the store metadat

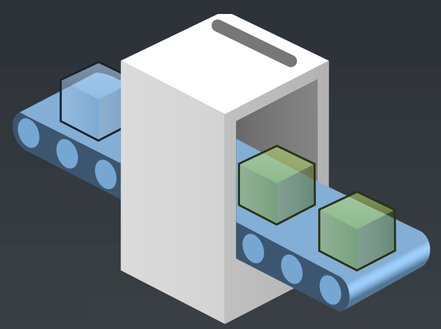


Demo

Creating a managed object model in Xcode

Adam Swift
Engineer, Core Data

http://developer.apple.com/wwdc2005/



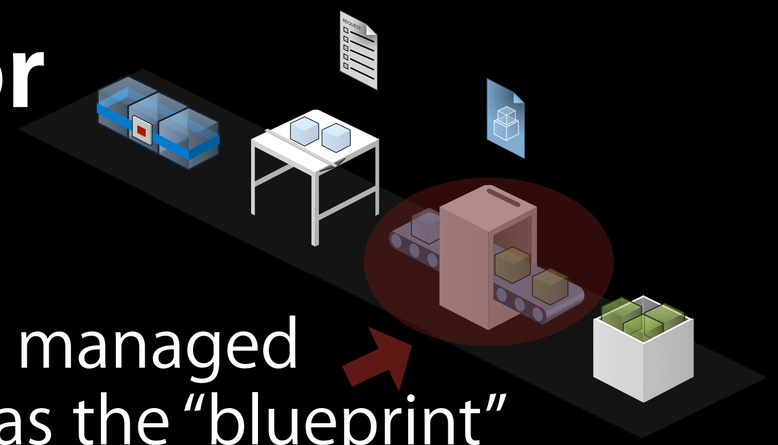
The Persistent Store Coordinator

The framework's middleman

http://developer.apple.com/wwdc2005/

The Persistent Store Coordinator

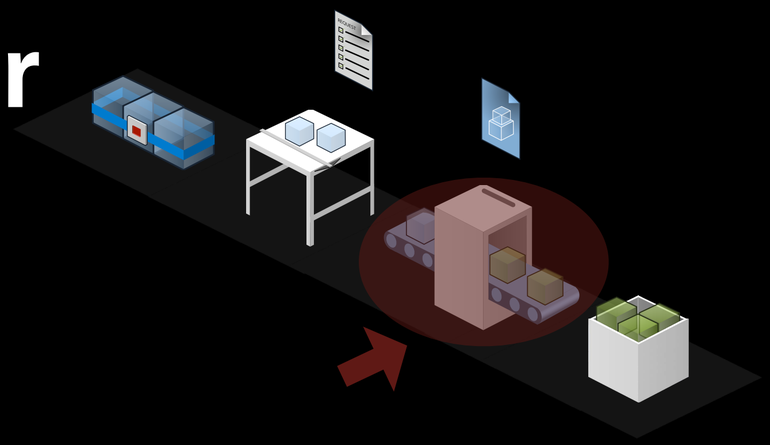
The framework's middleman

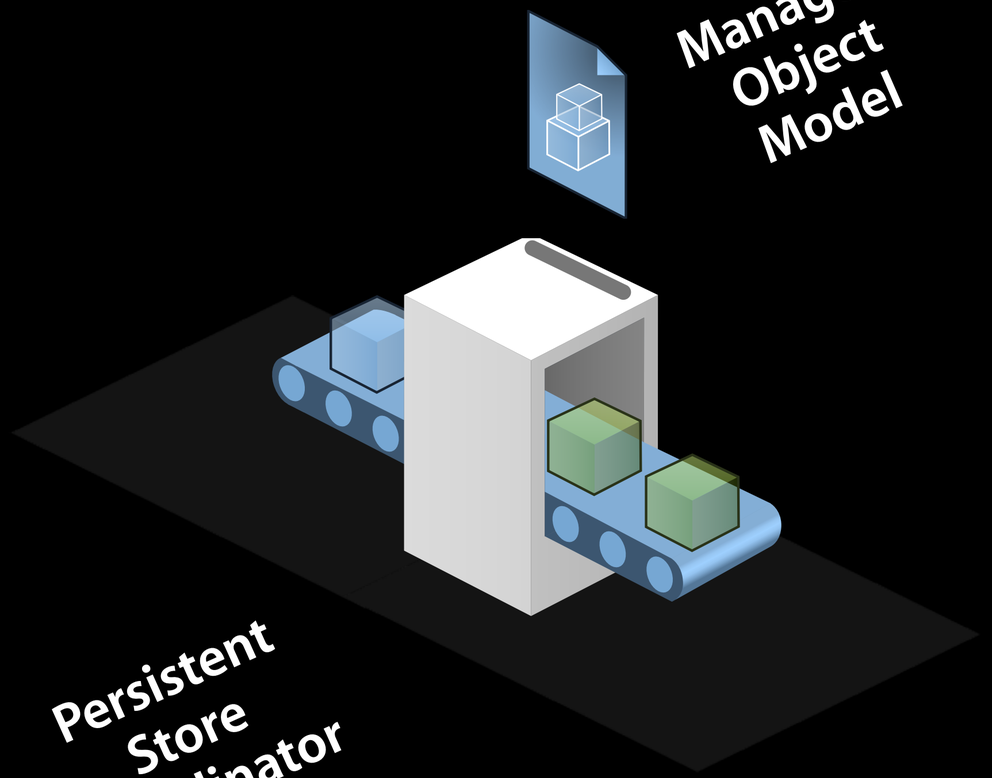


- Coordinates moving objects between stores and managed object contexts, using a managed object model as the "blueprint"
- Referred to commonly as the "PSC"
- Synchronizes access to persistent stores
- Unified representation for multiple stores
  - Striping, no mirroring

The Persistent Store Coordinator

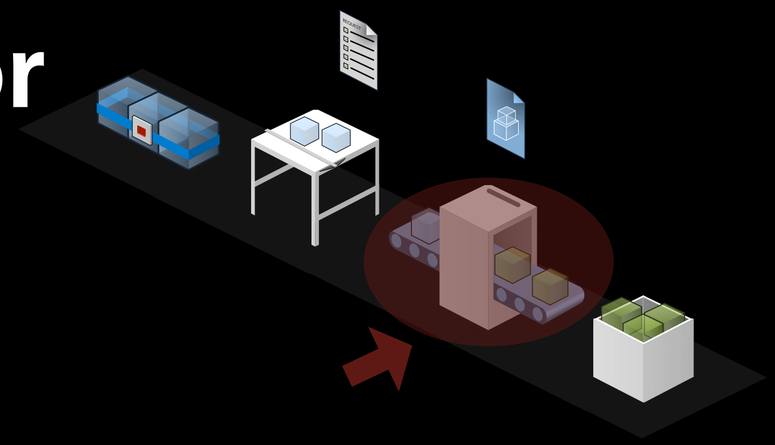
The facade in front of stores

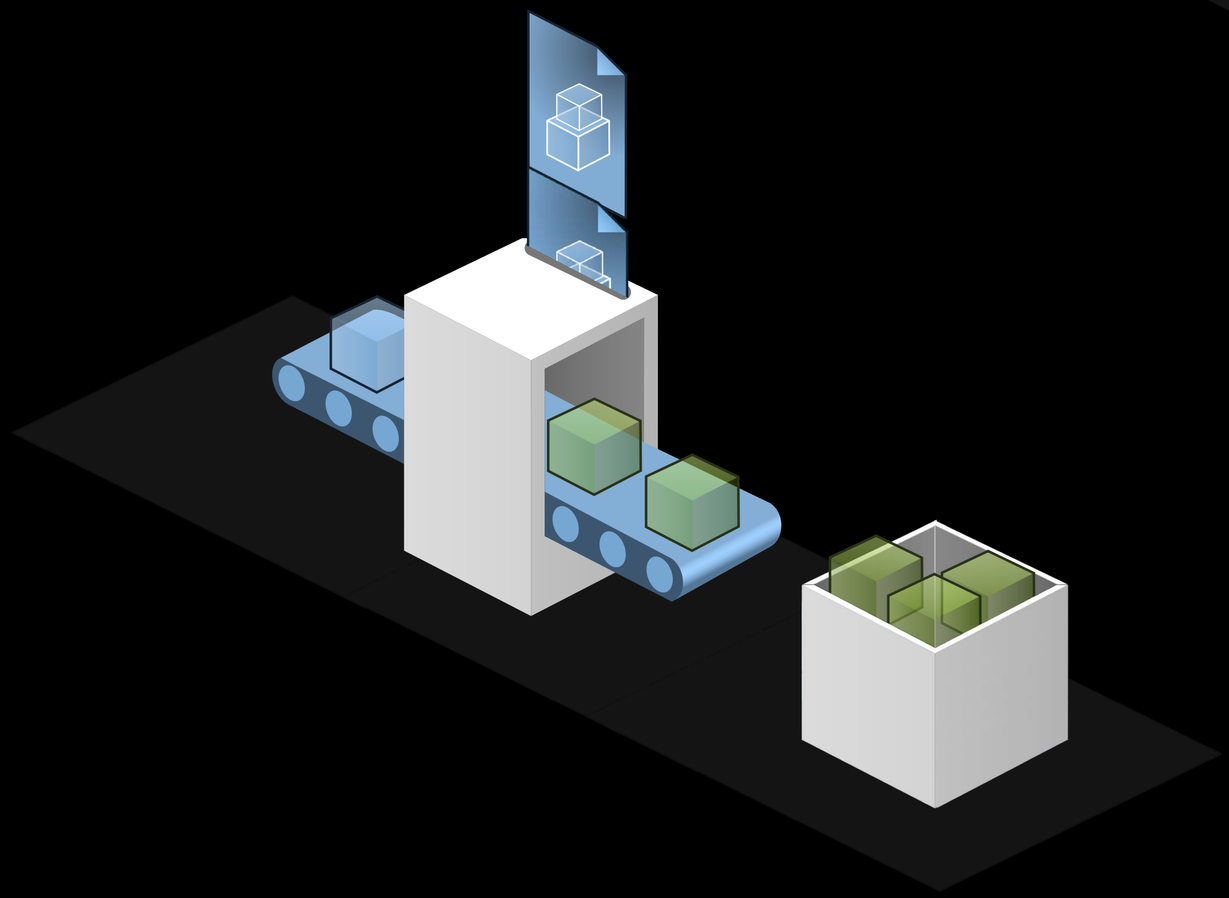




The Persistent Store Coordinator

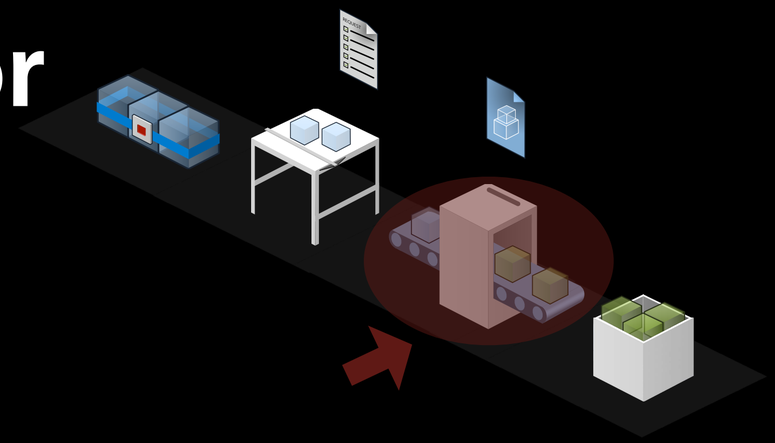
The facade in front of stores

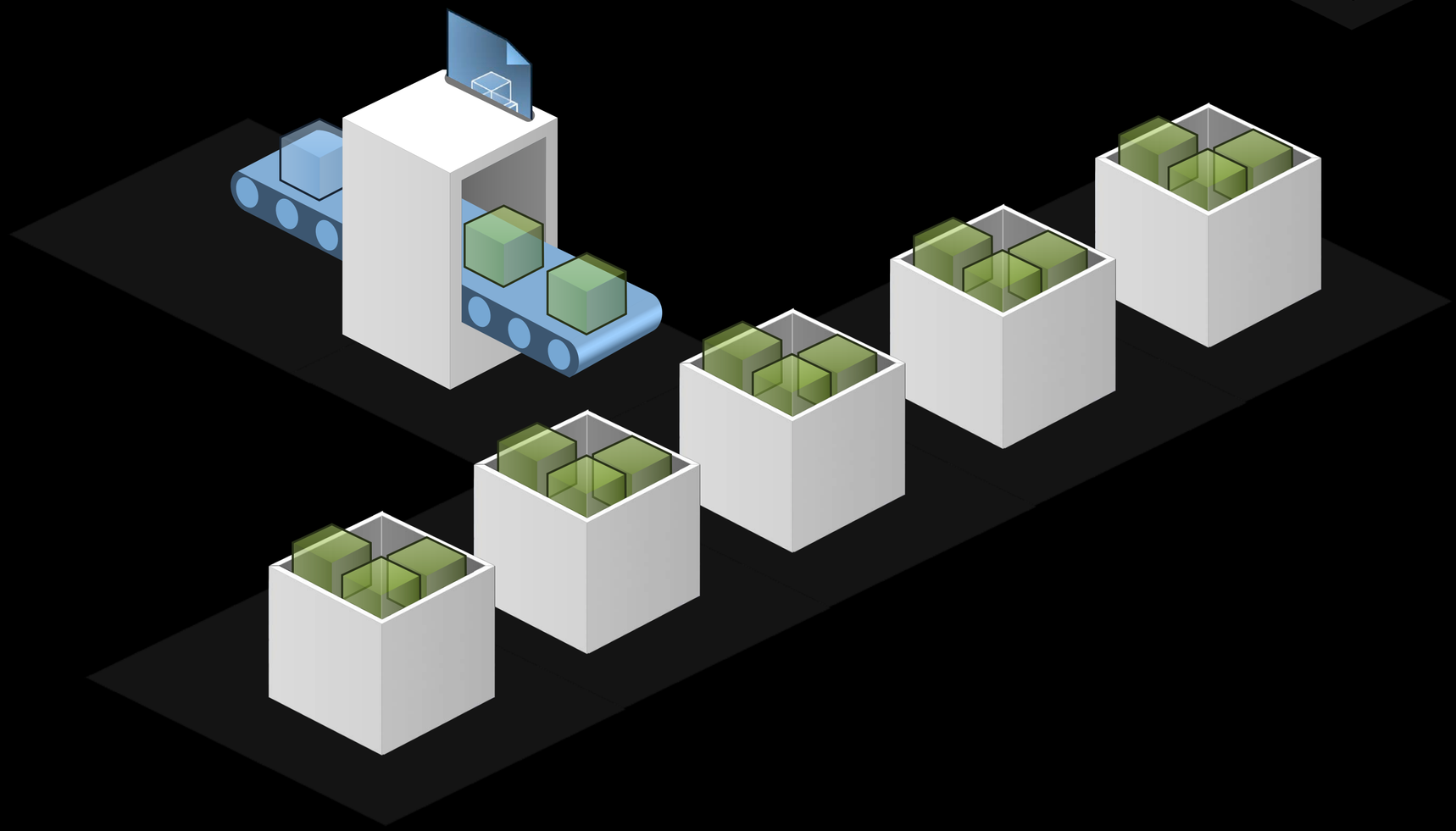




The Persistent Store Coordinator

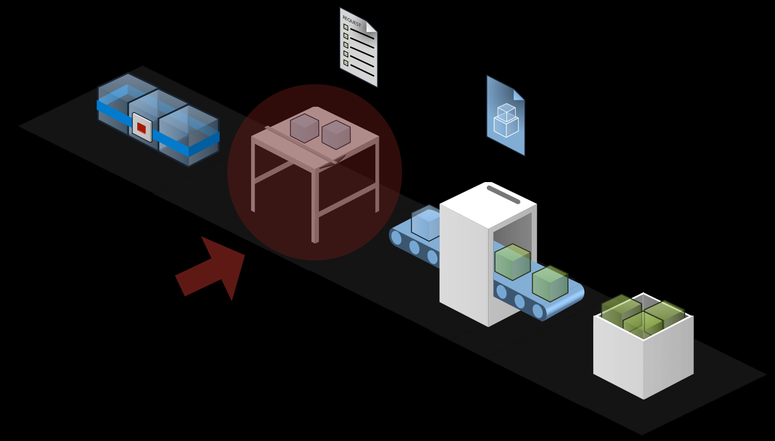
The facade in front of stores

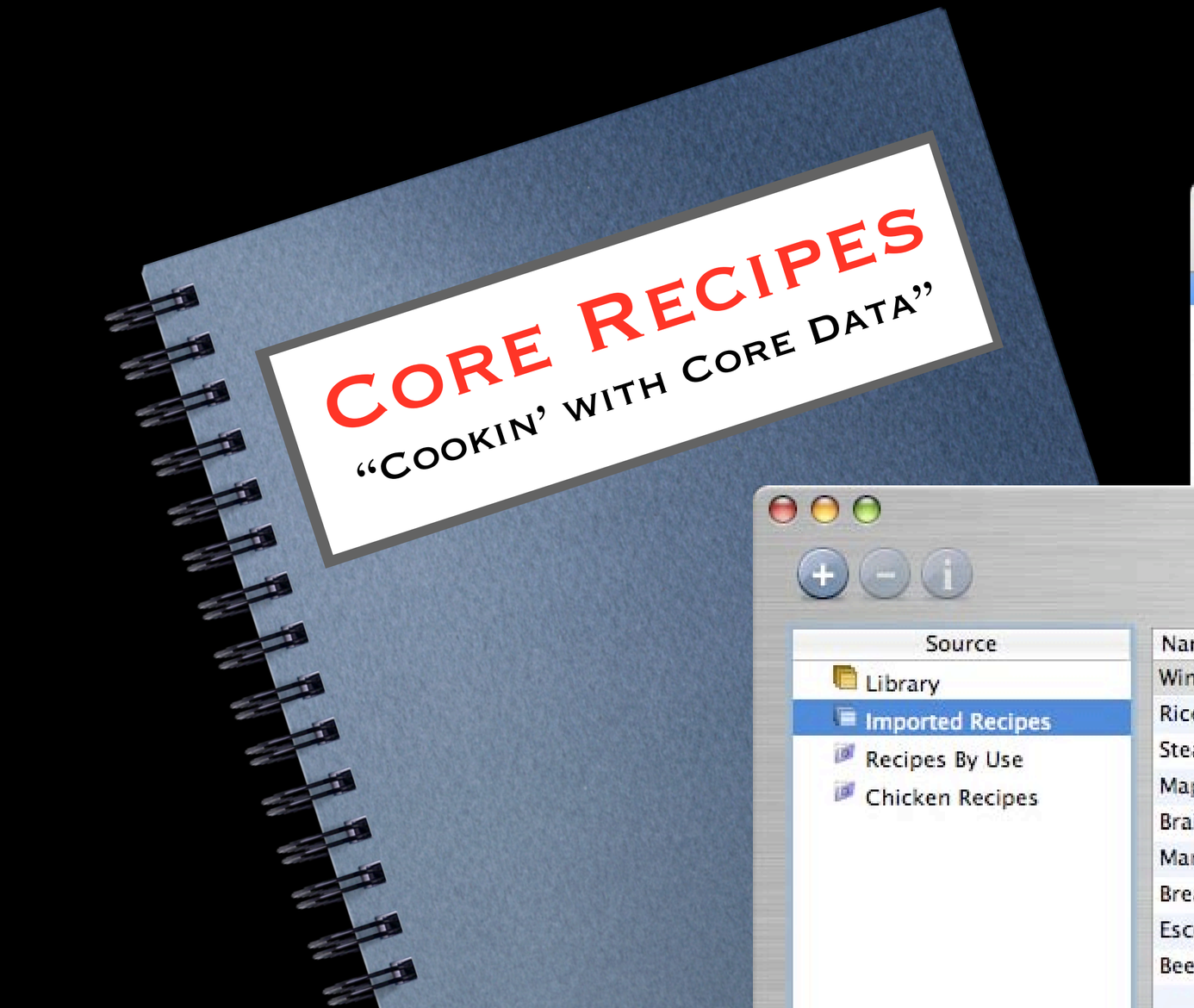


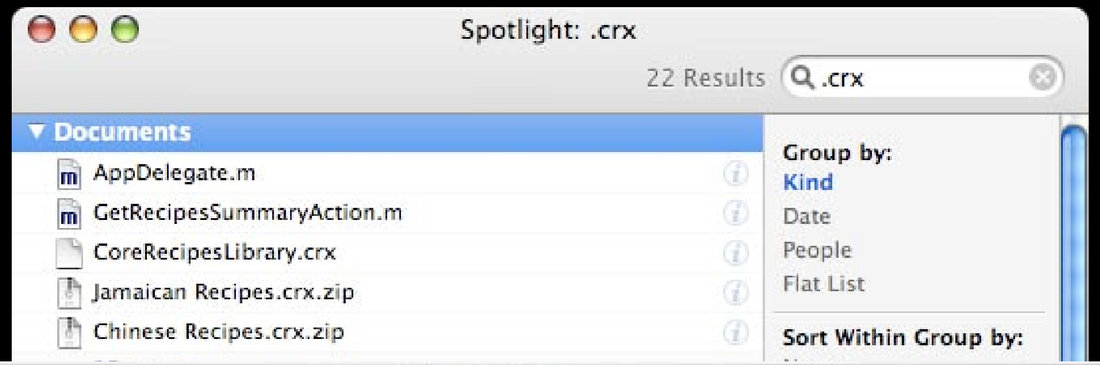


Core Recipes App

Uses Spotlight to find and load stores







What About "Metadata"?

How Core Data and Spotlight relate…

- Spotlight and Core Data are complementary:
  - Core Data is about the "inner workings" of your a
  - Spotlight is about an "outside view" of your conte
- The PSC provides simple API to read and write m tores

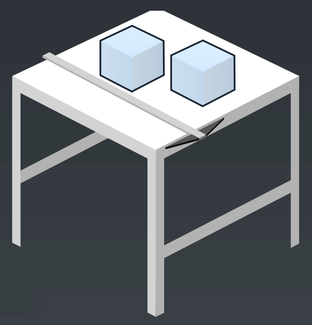
API Reminder:

```
NSPersistentStoreCoordinator
metadataForPersistentStore:
```

Sample Target and Code:

CoreRecipesApp

CoreRecipesSpotlightImporter



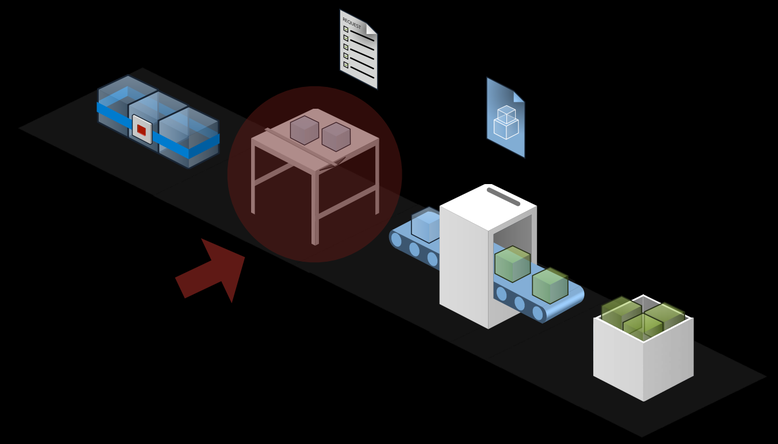
The Managed Object Context

The drafting-table for your data

http://developer.apple.com/wwdc2005/

The Managed Object Context

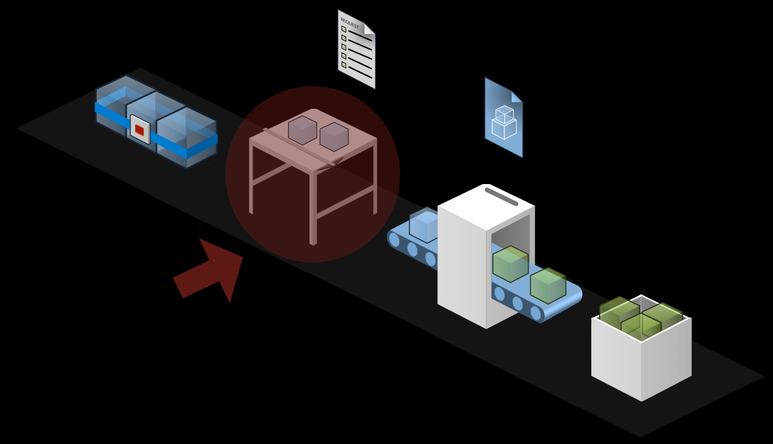
The drafting-table for your data

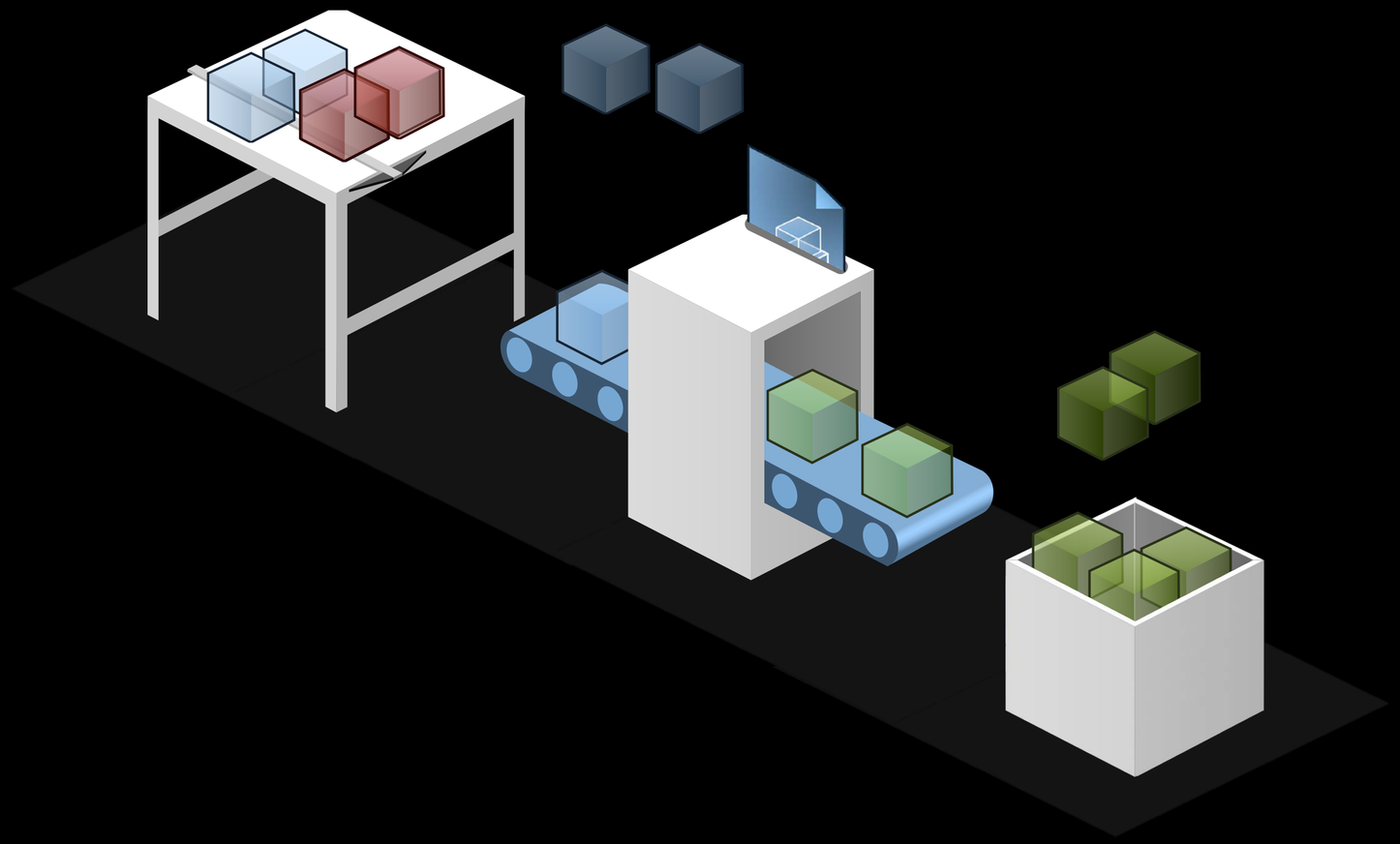


- Primary "workspace" of Core Data framework
  - Retrieve objects from stores
  - Making, saving, and reverting changes
  - Handling notifications of Core Data events
- Consider it a "scratchpad" for working with your data
  - Undo or revert changes at any time
- Referred to commonly as the "**MOC**"

The Managed Object Context

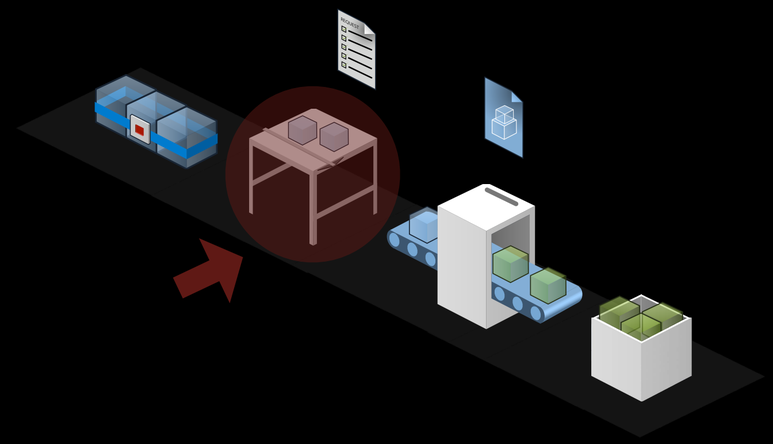
The drafting-table for your data

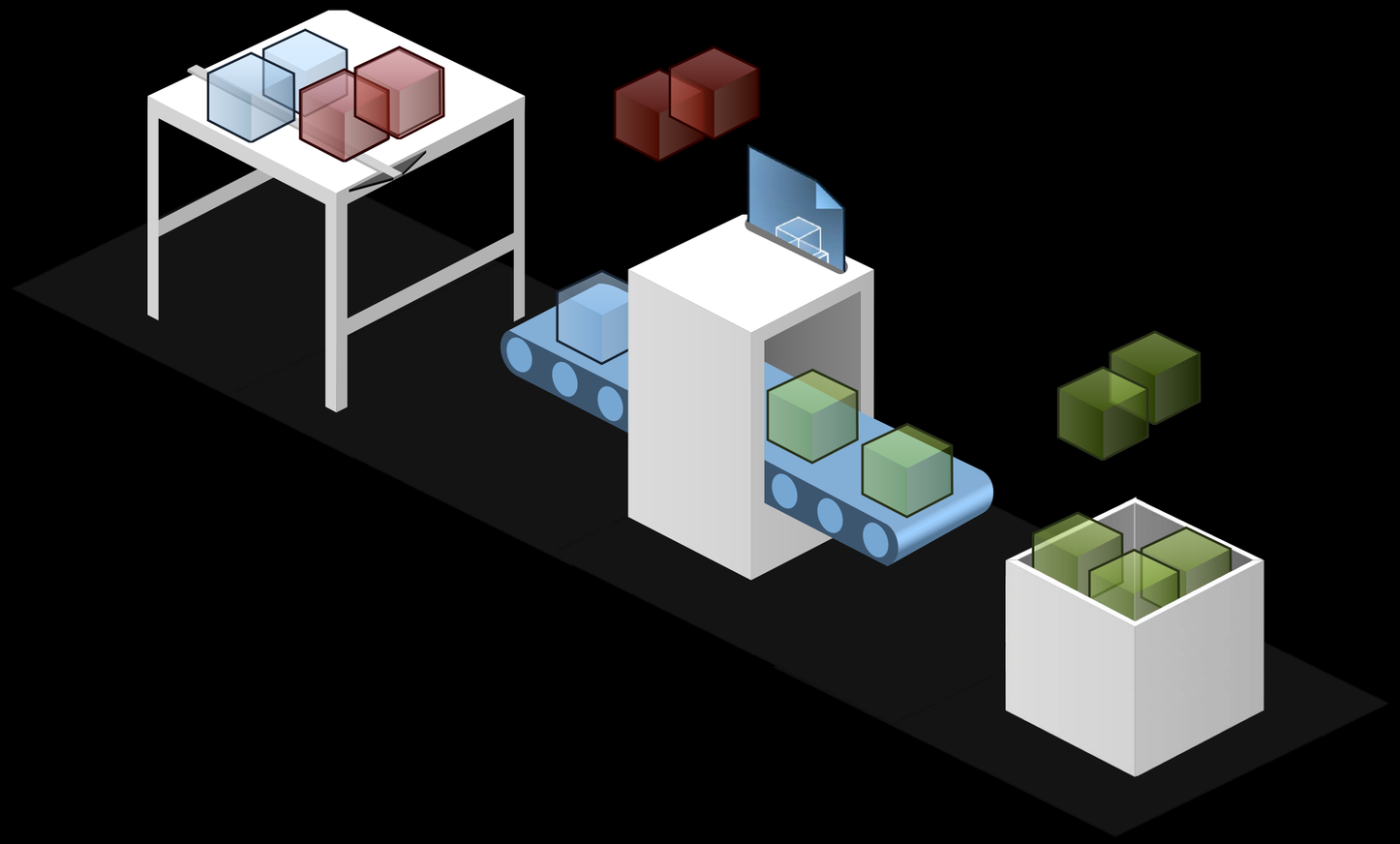




The Managed Object Context

The drafting-table for your data



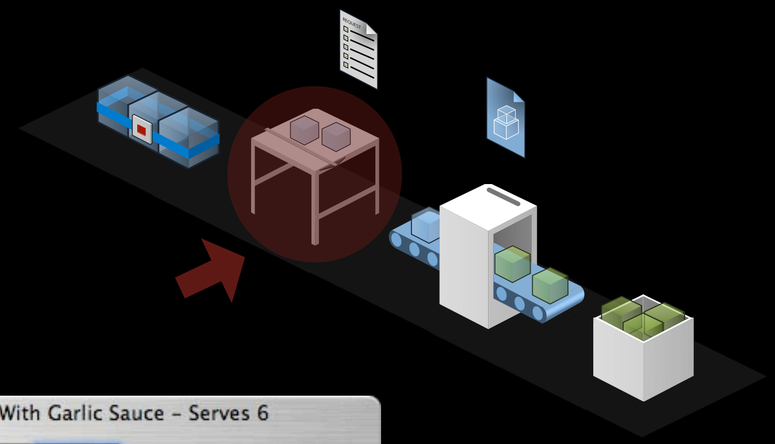


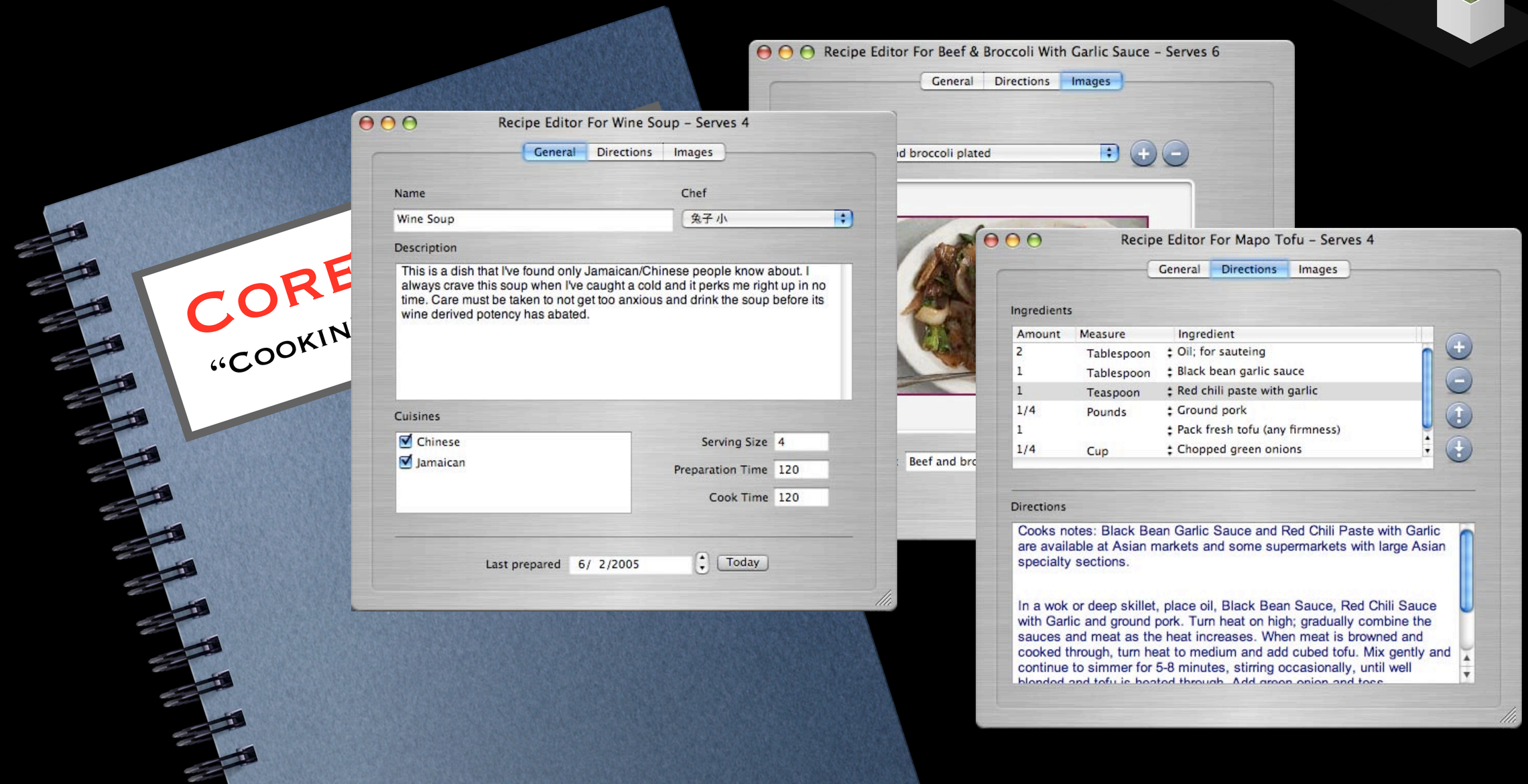
Many Edits, Many Contexts

Use multiple MOCs for your work

Core Recipes App

Uses a separate MOC per editor window





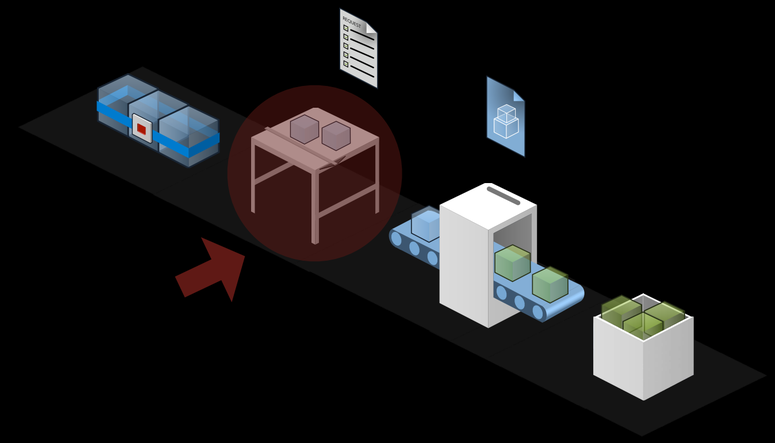
The Fetch Request

The bloodhound for your objects

http://developer.apple.com/wwdc2005/

The Fetch Request

The bloodhound for your objects



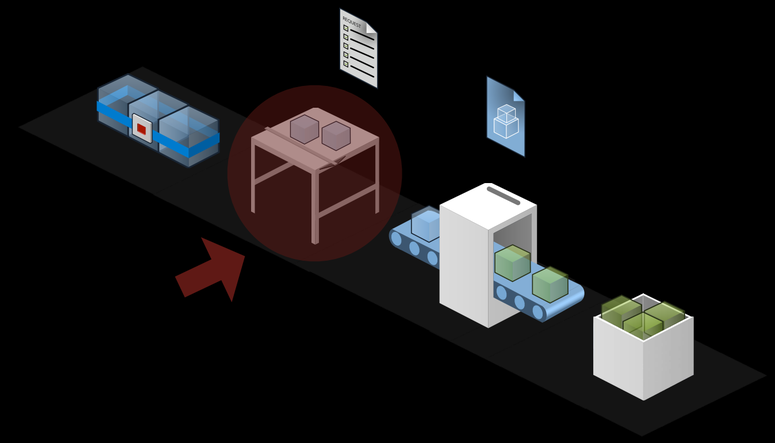
- An representation of a query operation
  - Specifies the entity to search on
  - A set of search criteria
  - Sort descriptors
- Fetch requests in models are named templates
  - Allows variable substitutions at runtime
- Evaluated in store where possible

Predicates

Defining criteria for your searches

- Predefined in model or created programmatica
- Often based on user input
- Executed in-memory or translated in persistent s
- Used outside of Core Data, too
  - Metadata API
  - Filtering in NSArrayController

Predicate Examples



Simple comparisons

```
name == “Tom”
```

Case/diacrytic insensitive lookups

```
name contains[cd] “Tom”
```

Logical operations

```
(firstName contains “a”) OR (lastName contains “a”)
```

Predicates for relationships

```
group.name matches “work*”
ALL children.age > 12
ANY children.age > 12
```

Operations

```
@sum.items.price < 1000
```

Core Recipes App

Predicates, predicates, everywhere

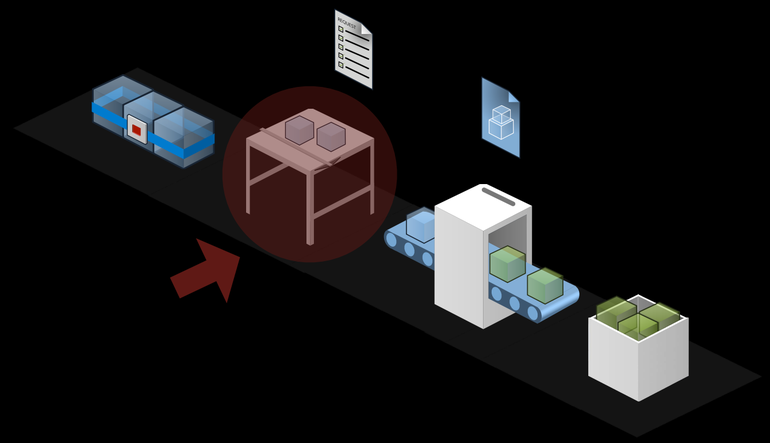
- Filter field uses compound predicates
- The “SmartGroup” entity heavily uses predicates
- Uses a predicate to find stores via Spotlight
- Has an inference engine based on predicates

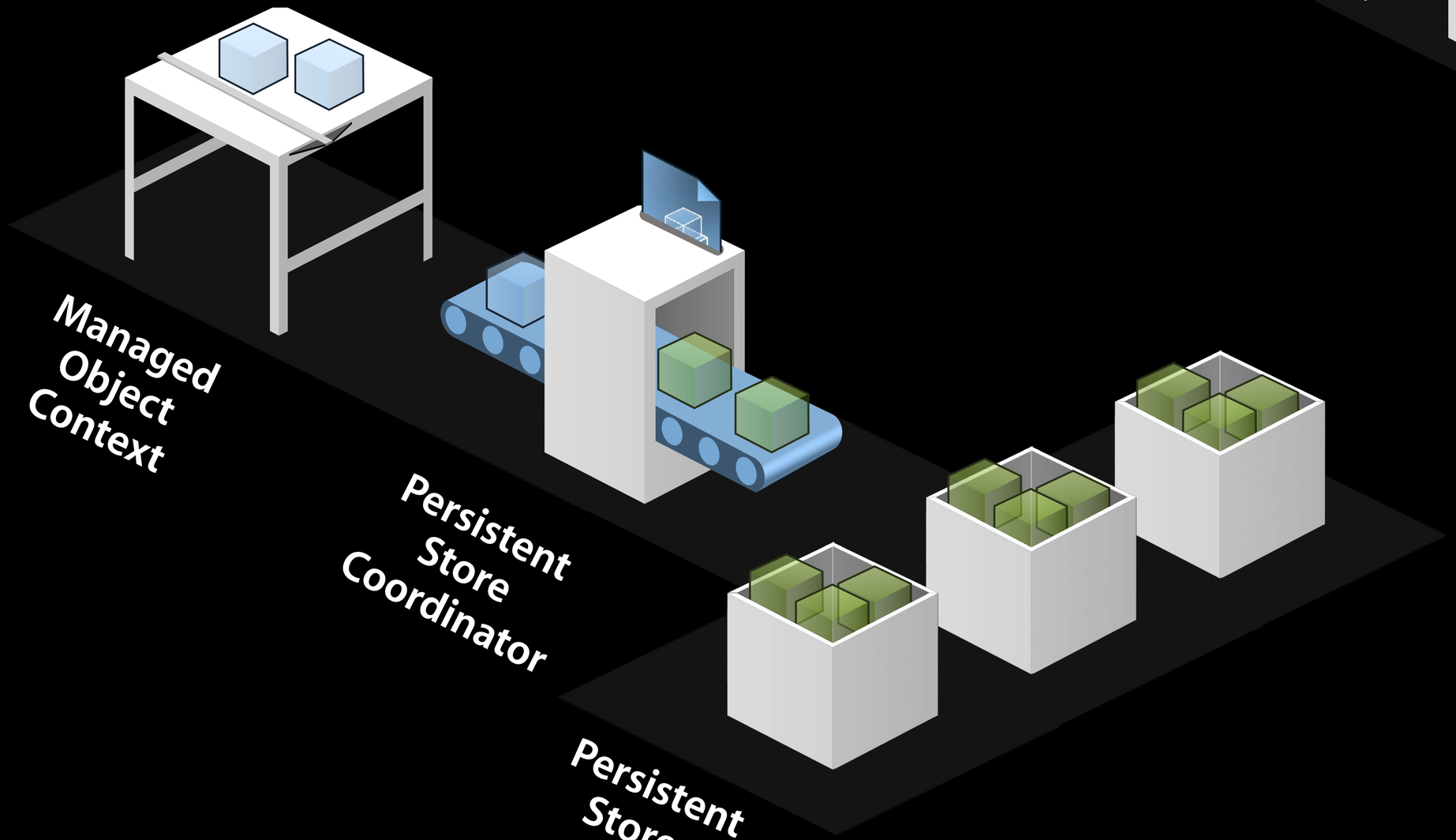
Code Snippet
(from CoreRecipesApp):

```
AppDelegate.m
InferenceCore.m
```

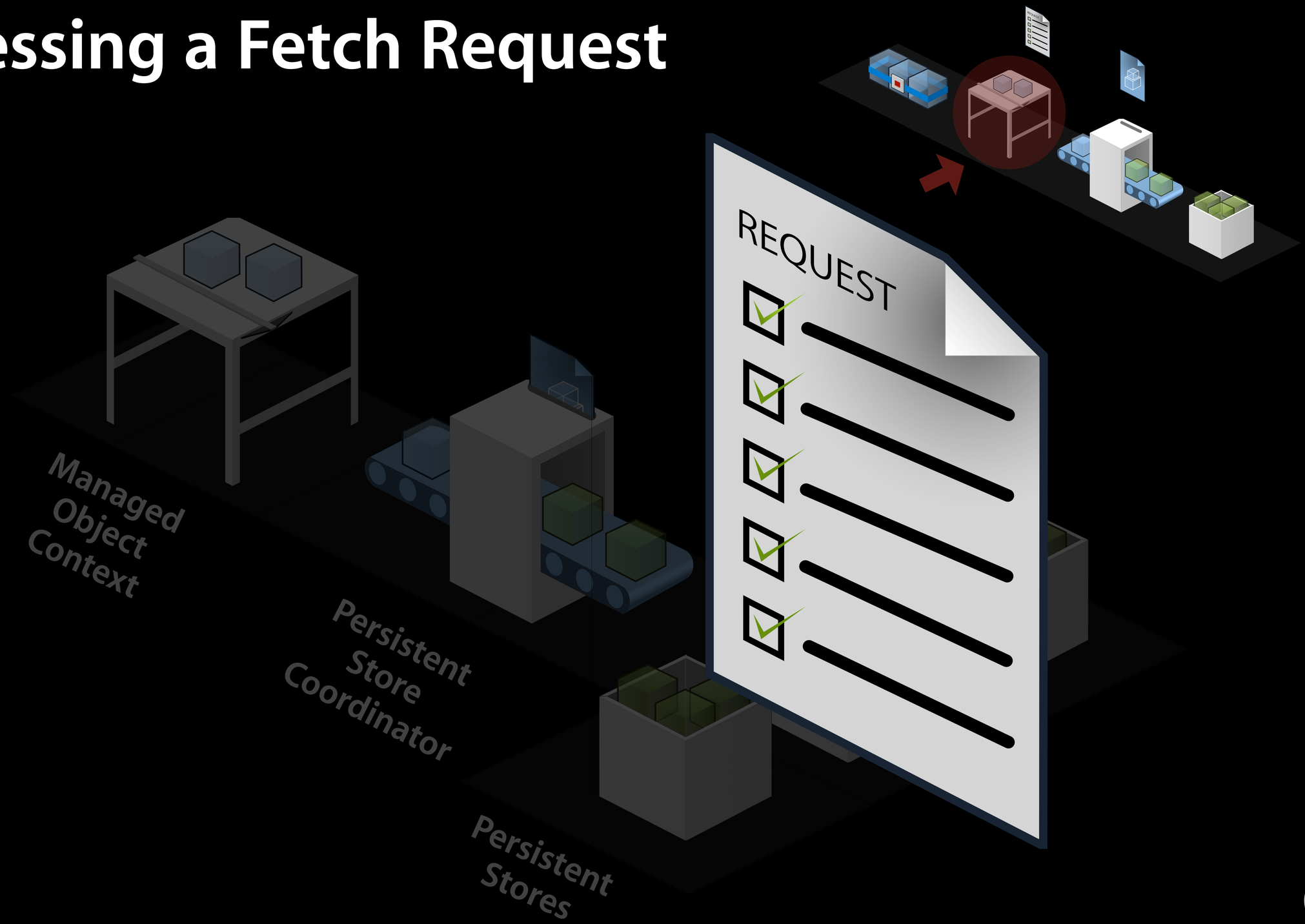
Processing a Fetch Request

Follow the bouncing ball

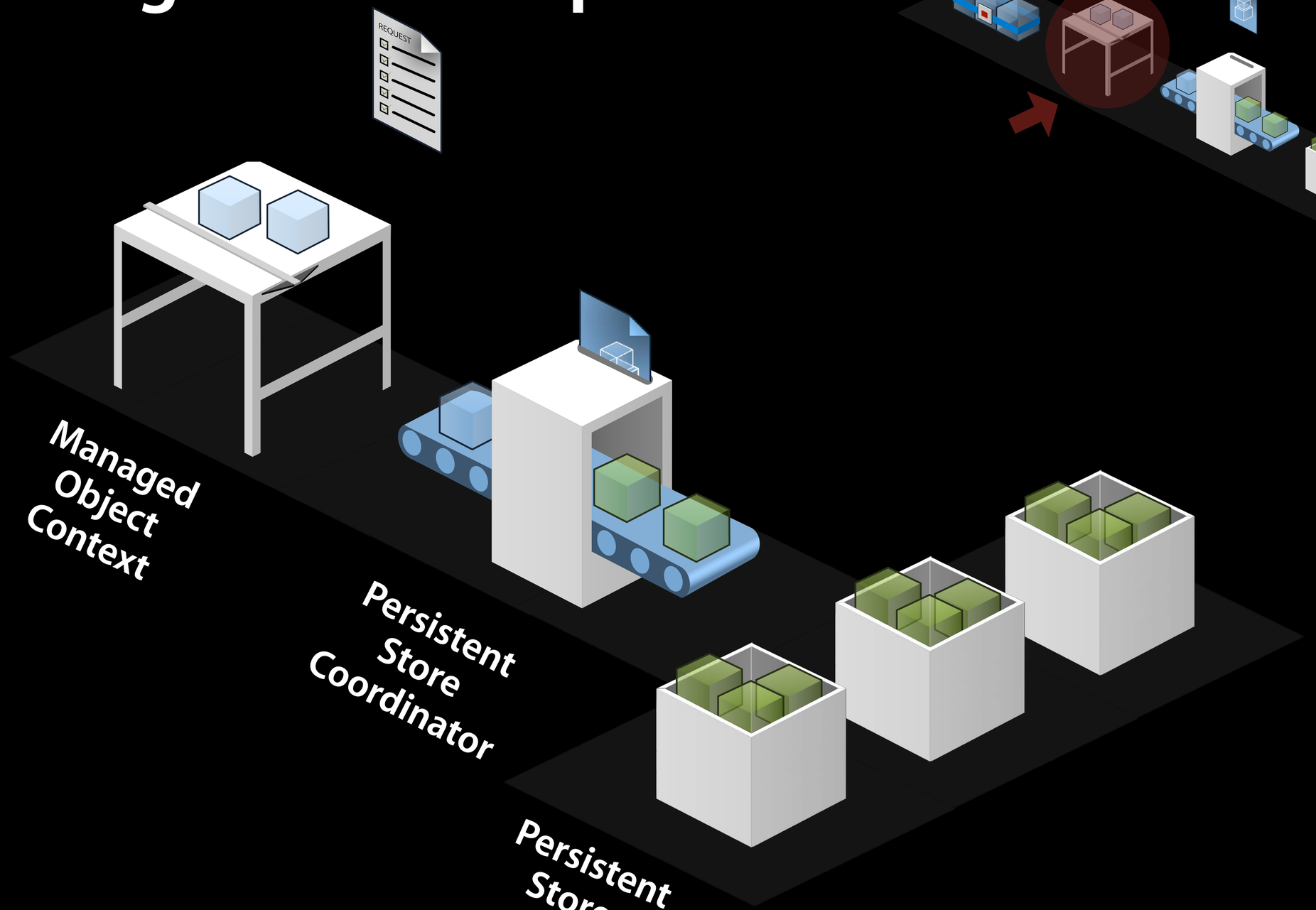


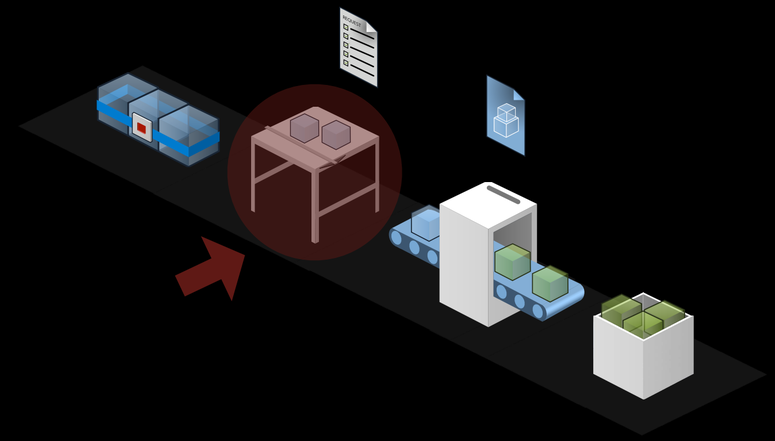


Processing a Fetch Request

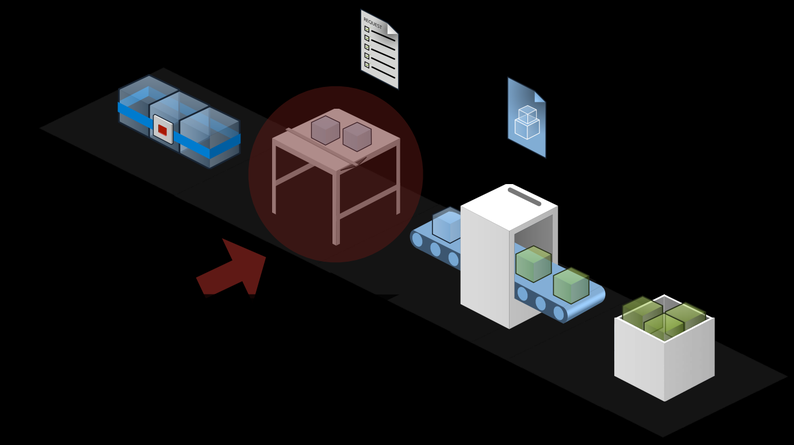


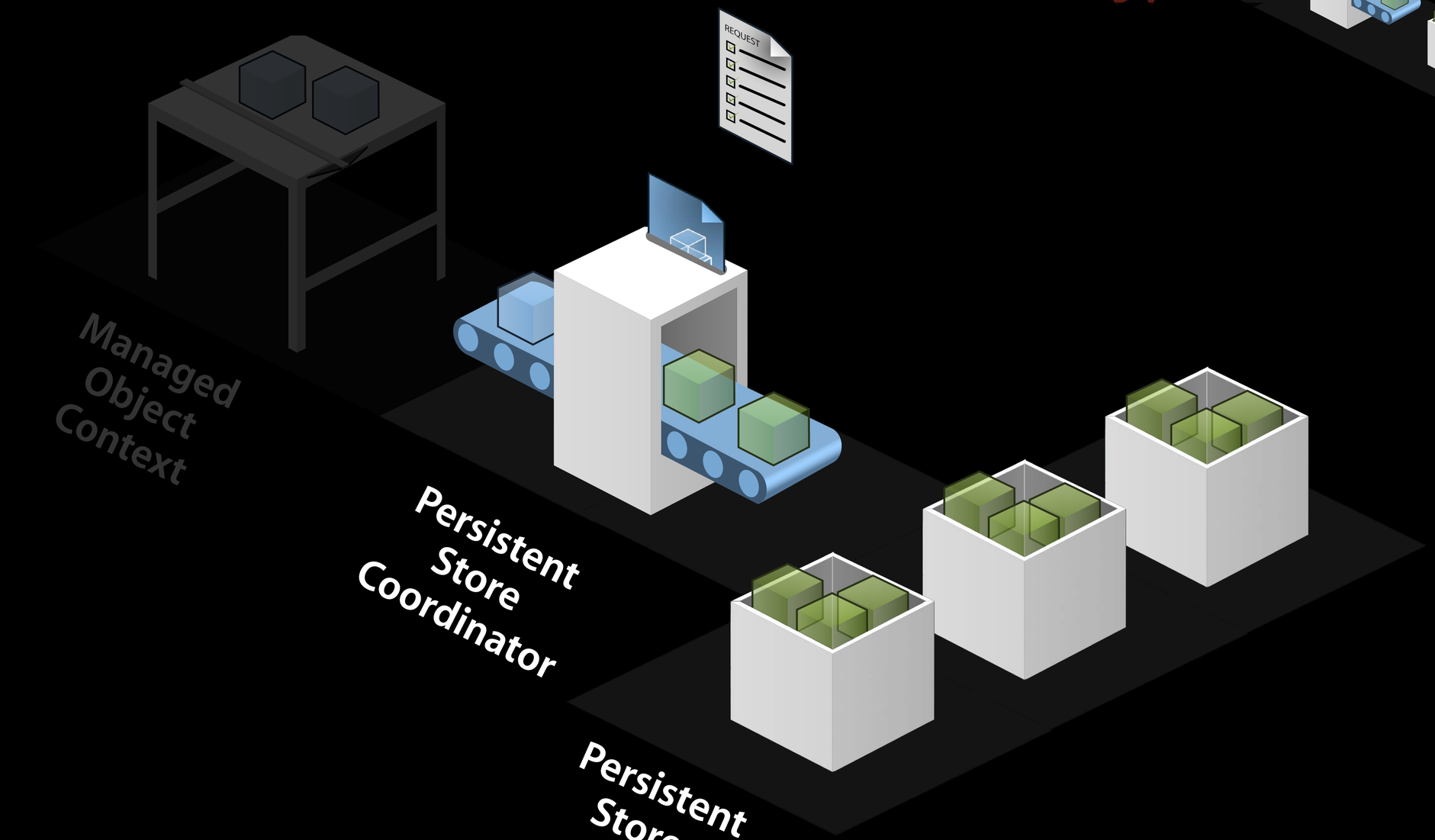
Processing a Fetch Request



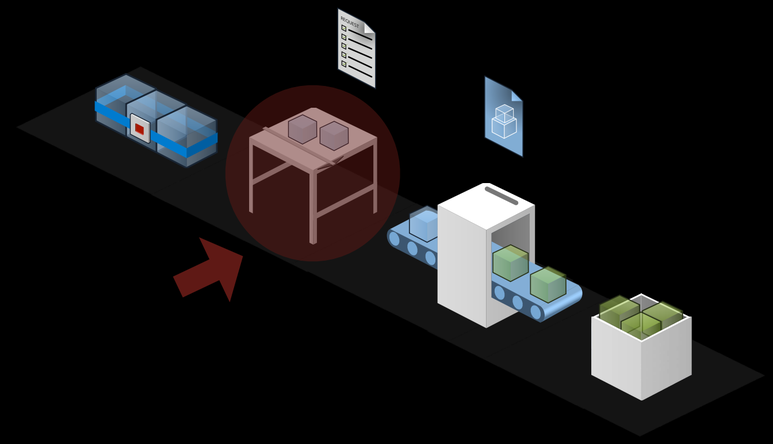


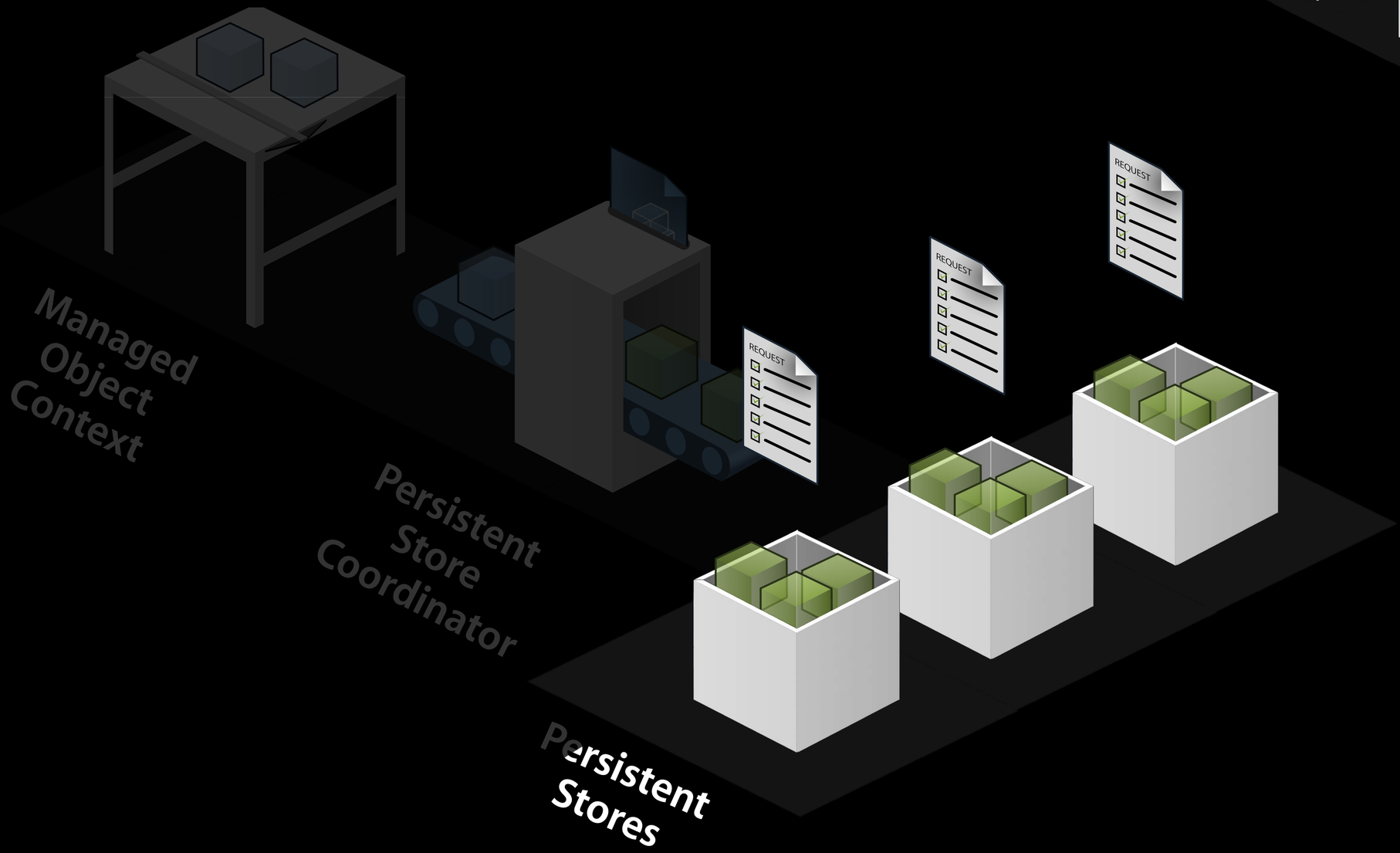
Processing a Fetch Request



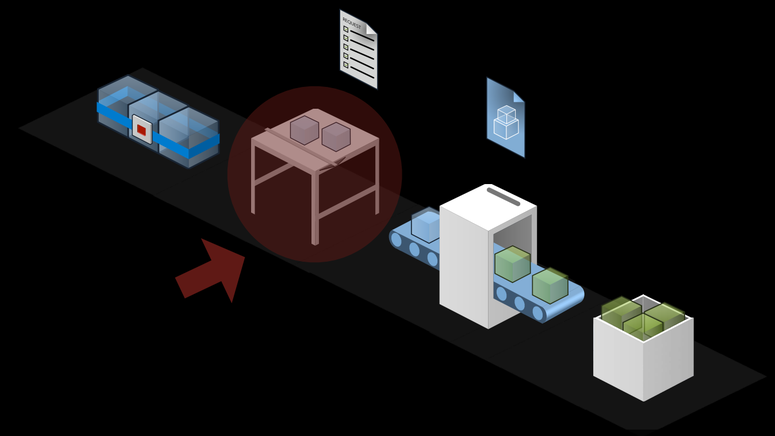


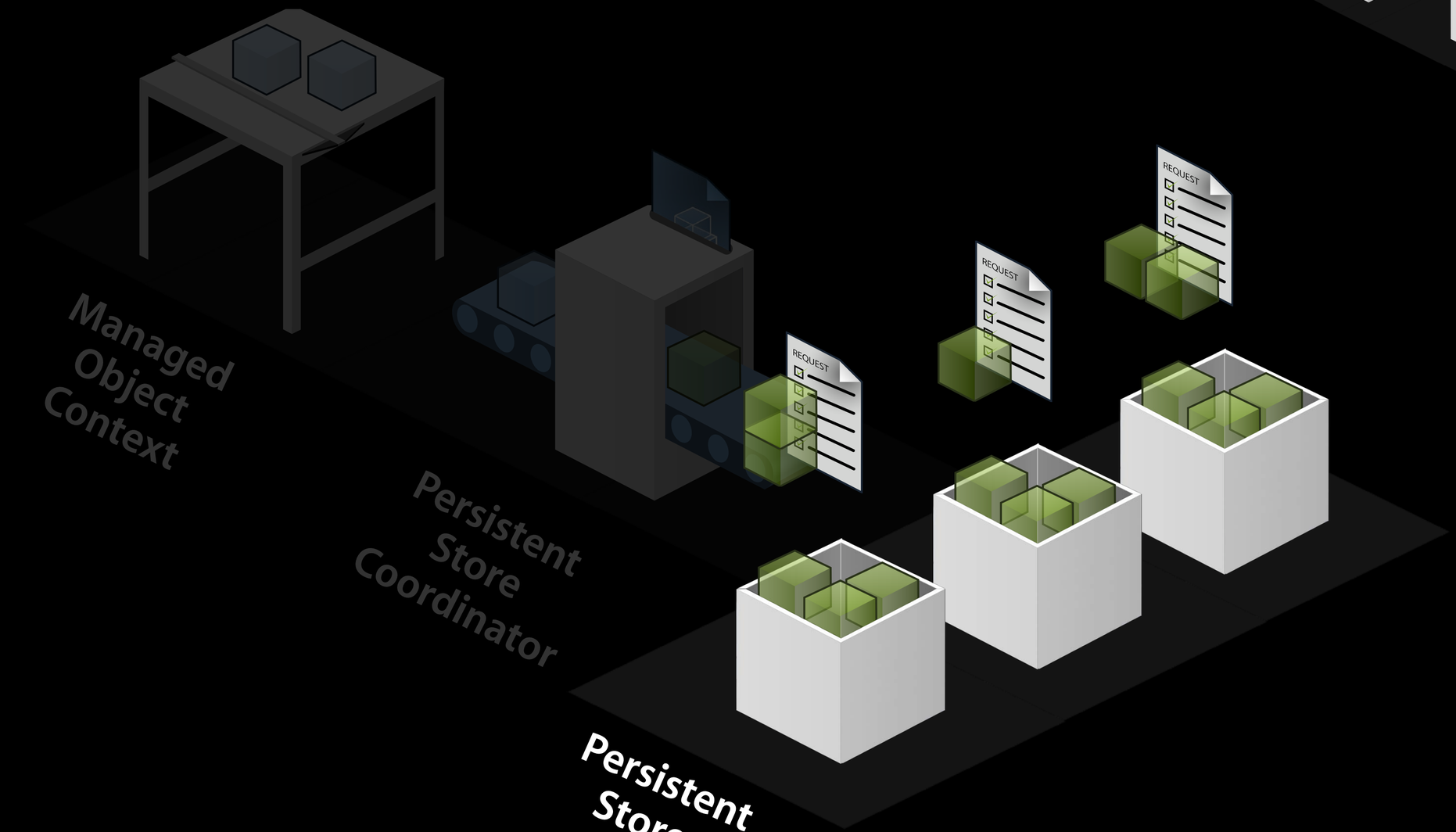
Processing a Fetch Request



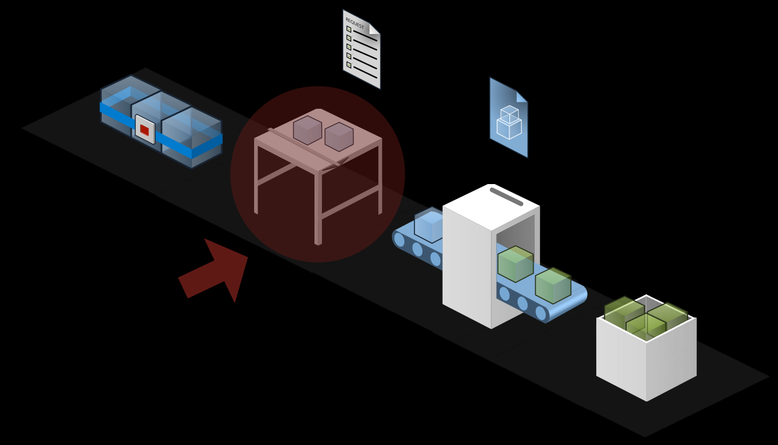


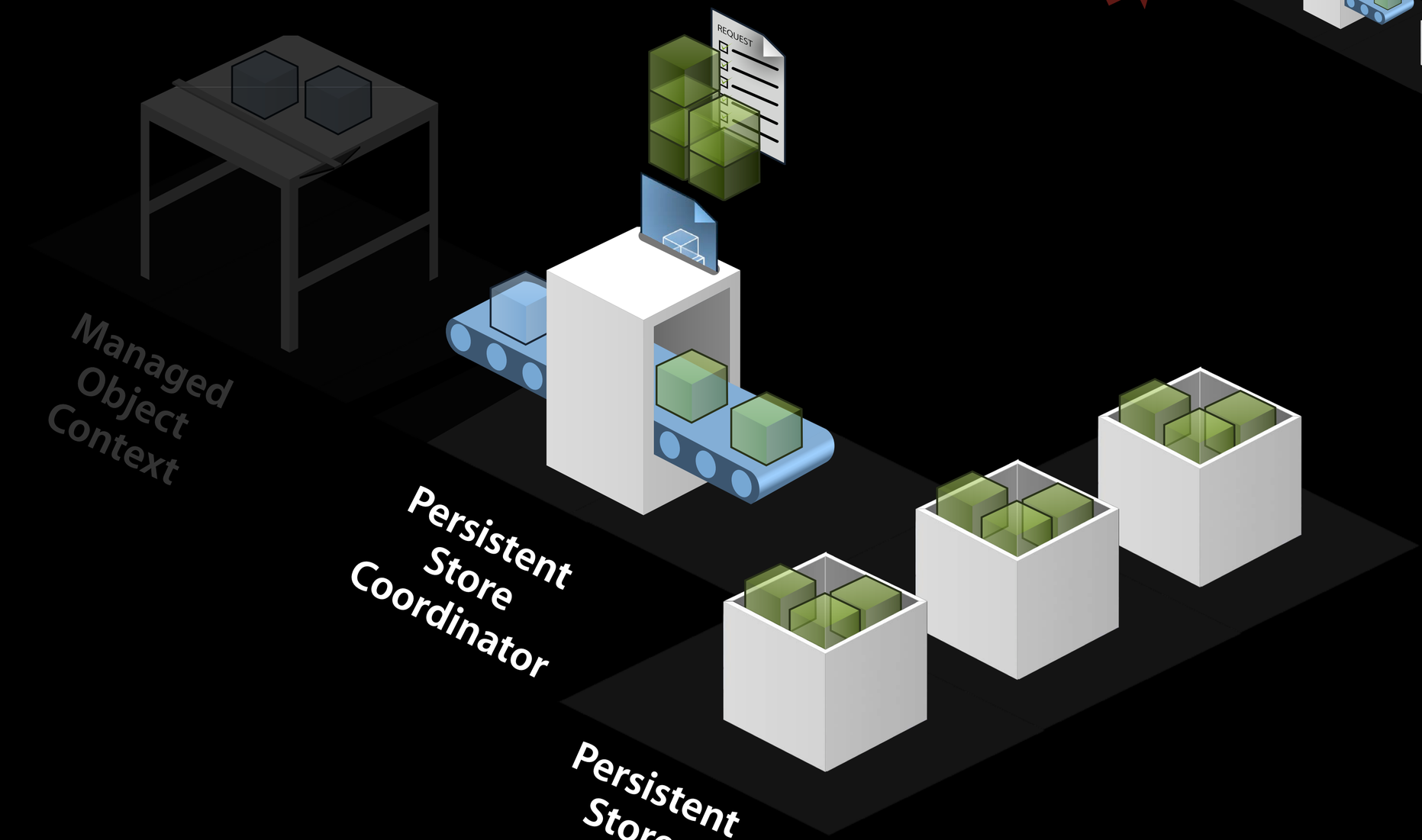
Processing a Fetch Request



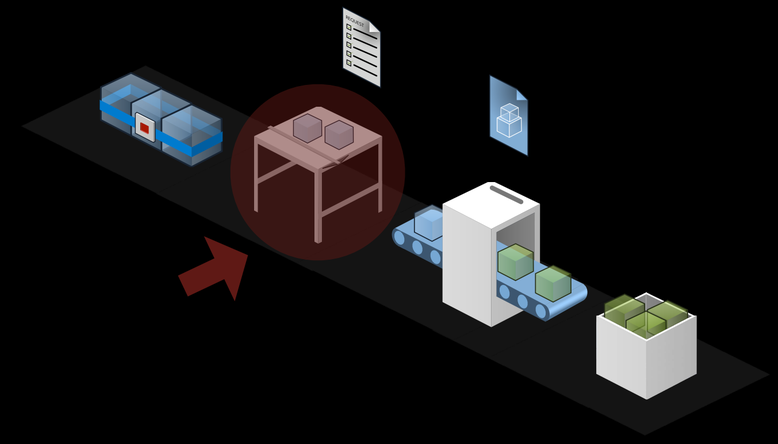


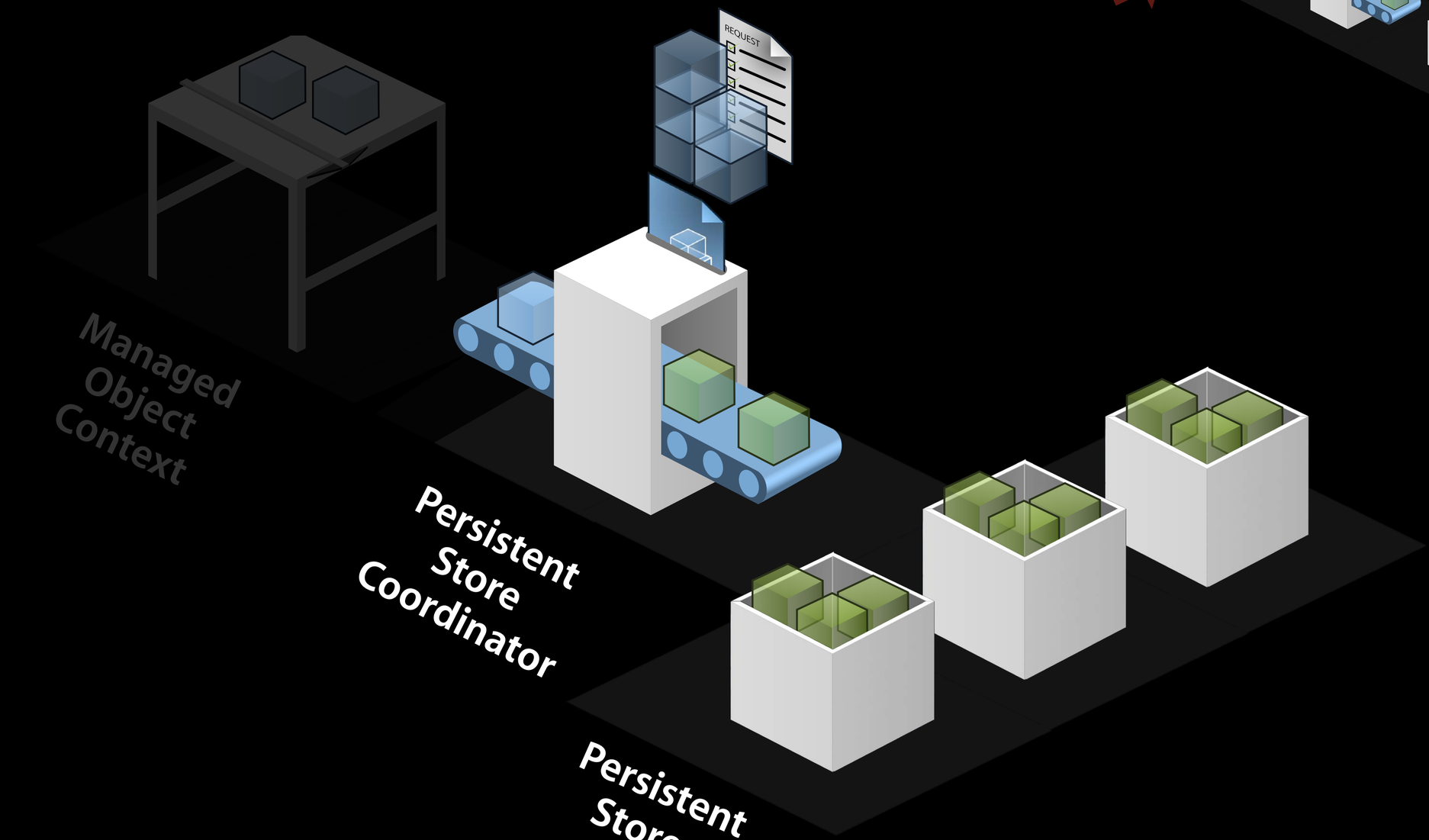
Processing a Fetch Request



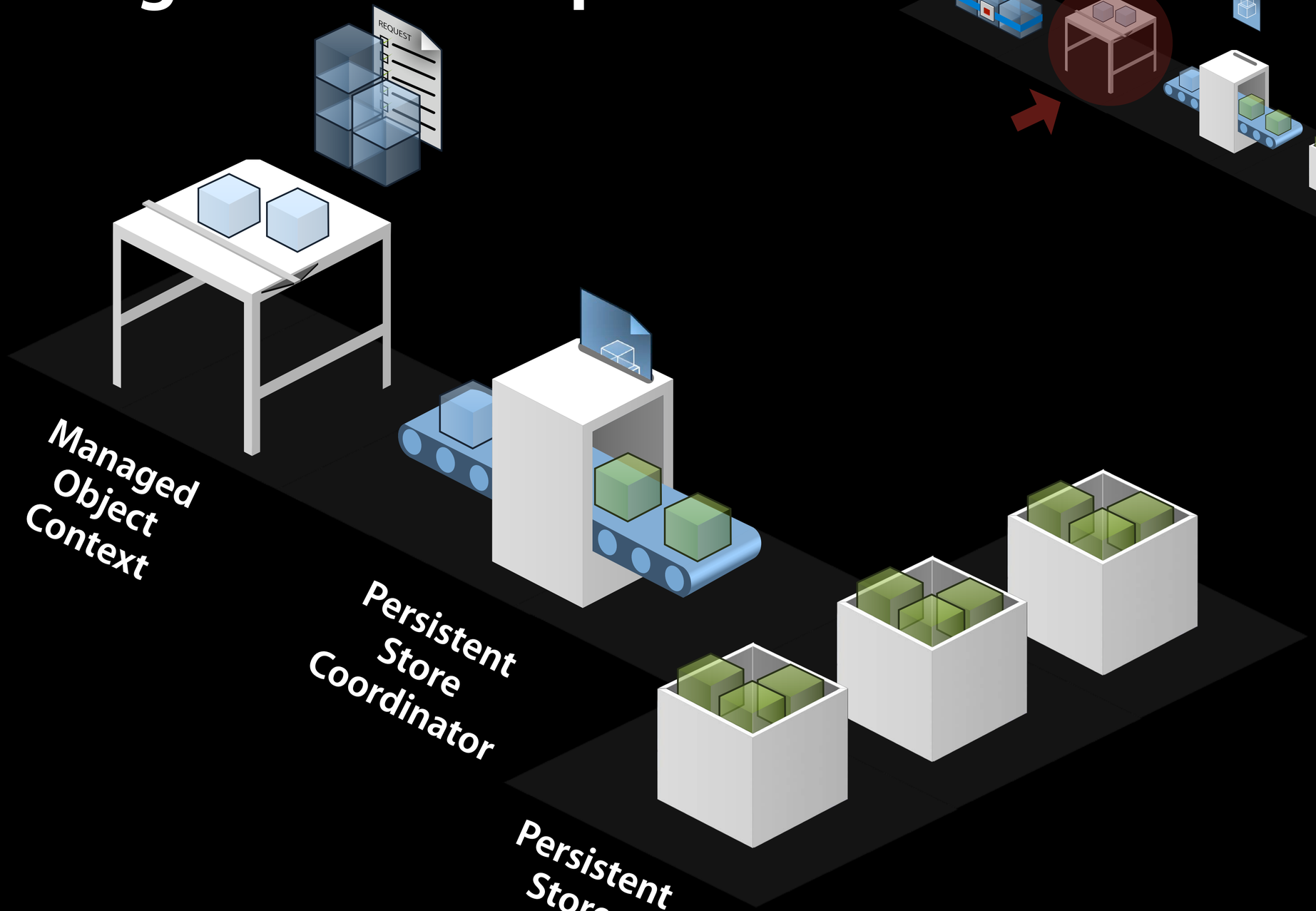


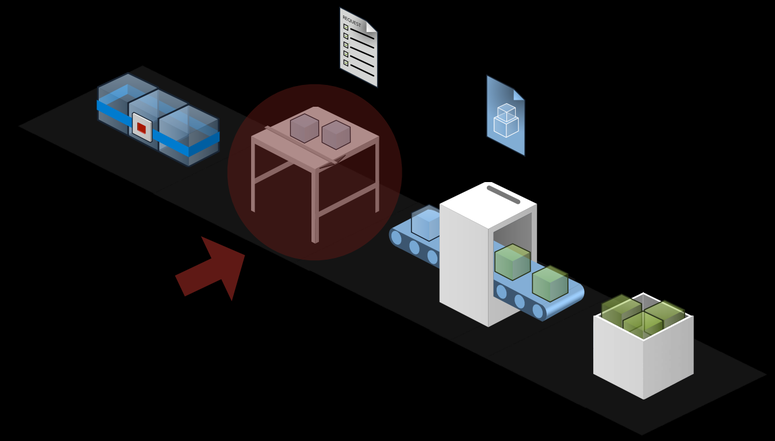
Processing a Fetch Request

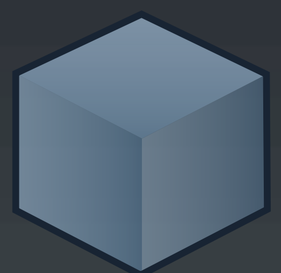




Processing a Fetch Request







The Managed Object

Your data workhorse

http://developer.apple.com/wwdc2005/

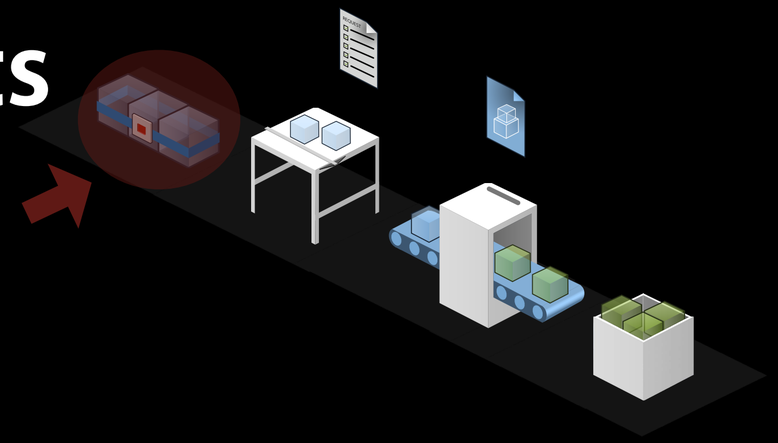
The Managed Object

Your data workhorse

- Basic unit for development and operation in Core
  - Fetch, insert, update, delete, save
- Each managed object
  - Is described by a single entity
  - Is associated with a single context
  - Has a unique ID
- Core Data provides a basic class
  - Conforms to KeyValueCoding
  - Conforms to KeyValueObserving

Making the Most of Your Objects

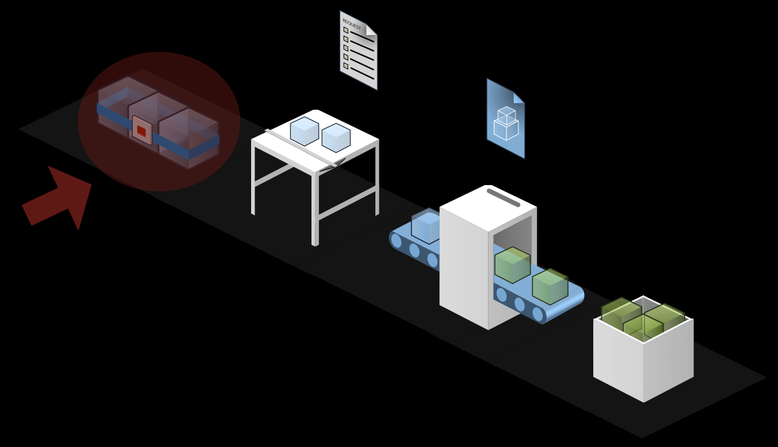
Options for adding "smarts"



- Consider where to provide functionality
  - Entity inheritance
  - Custom base class
  - Custom classes
- Use custom managed object (sub)classes for
  - Working with derived values
  - Reacting to state changes
  - Validation

Managed Object URIs

How to refer to items later



- Each managed object has a Uniform Resource Identifier (URIs) accessible via the object ID
- Can be used to refer to the object later
- Note: the URI for a temporary (unsaved) object is also temporary

Code Snippet:

CoreRecipesApp
RecipeUtilities.m
applicationURLForRecipe:

Demo

Creating a sample Core Data application

Adam Swift
Engineer, Core Data

http://developer.apple.com/wwdc2005/

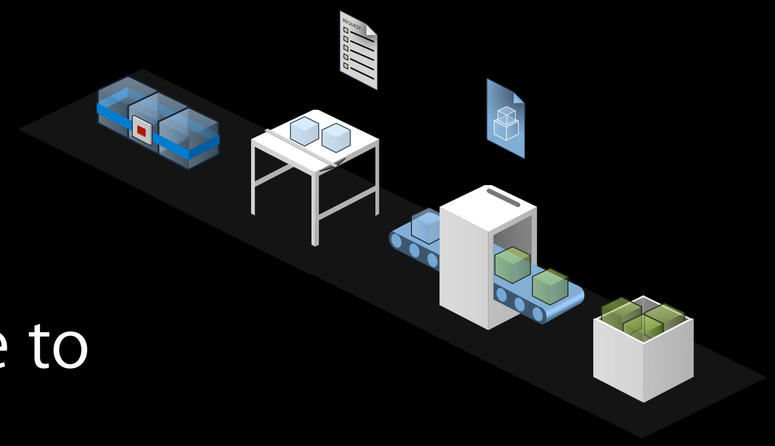
Two Common Questions

Things you might be wondering

http://developer.apple.com/wwdc2005/

Will It Be Fast Enough?

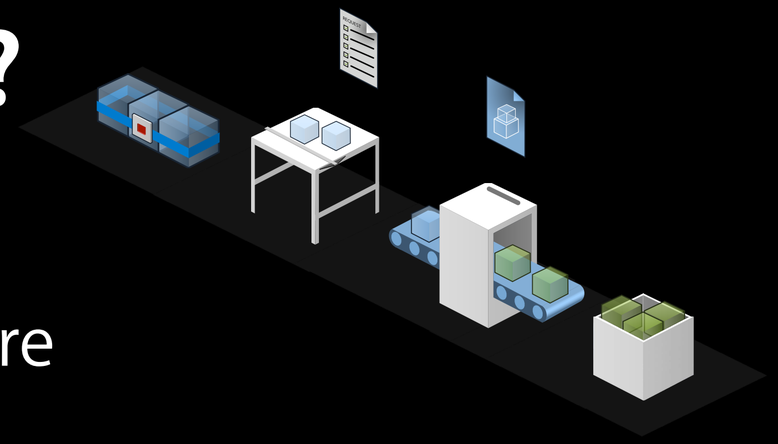
The need for speed



- In general performance tests, Core Data was able to fetch 30,000 objects in less than one second
- Put into perspective, that's more than the average
  - Number of songs in iTunes
  - Number of addresses in AddressBook
  - Number of messages in Mail

Will It Require a Lot of Memory?

My machine has a lot, but my users' don't



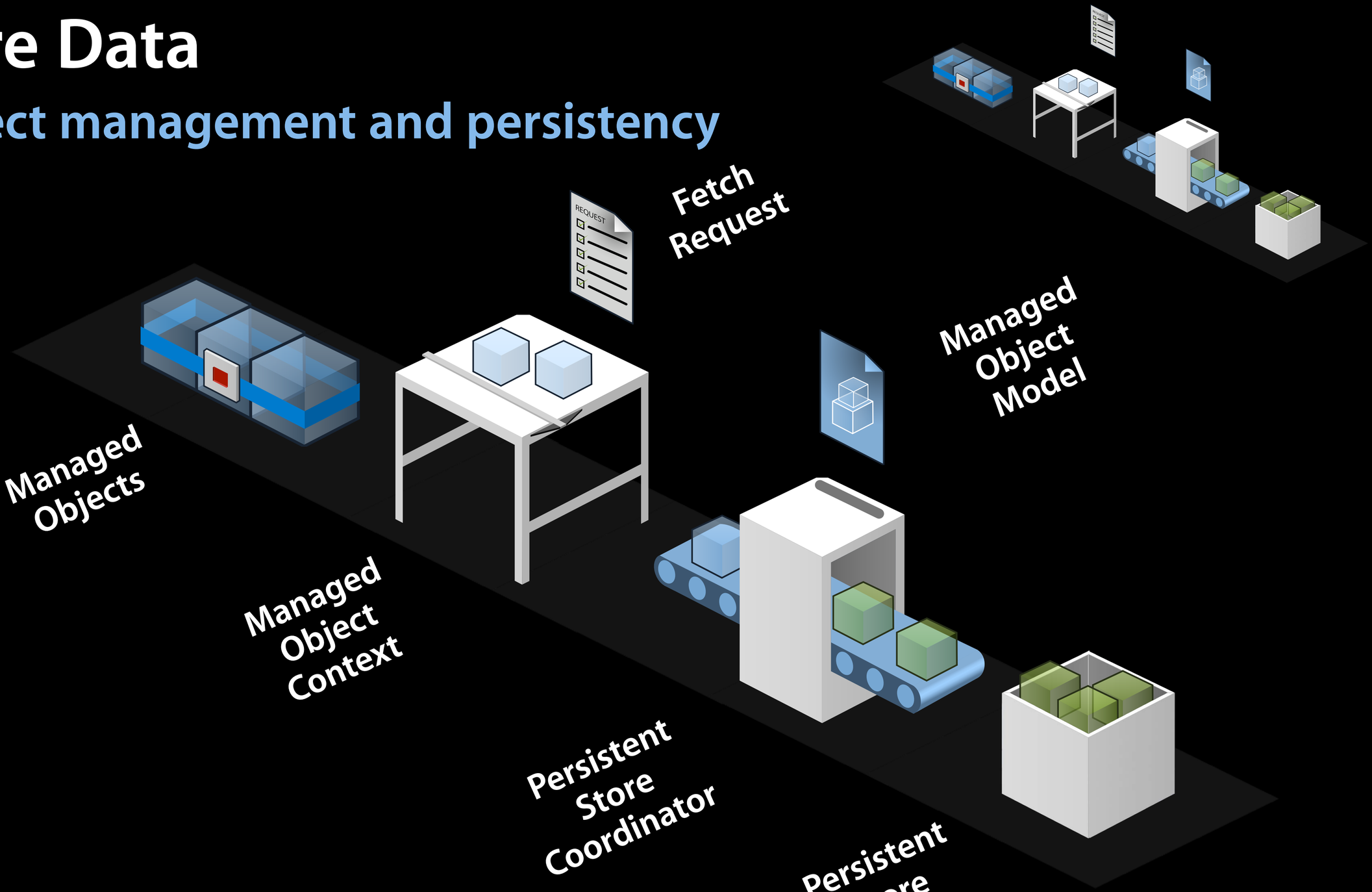
- The framework only keeps around objects that are needed in memory
  - Releases objects no longer in use
  - Grows the graph on demand
- The framework only creates objects when necessary
  - Objects from the store are "faults" until utilized
  - There is one underlying "cache" for objects

Summary

http://developer.apple.com/wwdc2005/

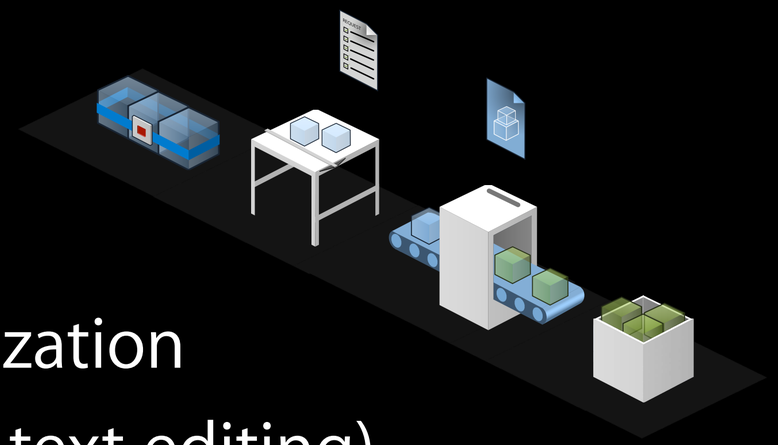
Core Data

Object management and persistency



Core Data

Object management and persistency



- Integration with Cocoa bindings for UI synchronization
- Built-in management of undo and redo (beyond text editing)
- Automatic validation of property values
- Grouping, filtering, and organizing data
- Change propagation, including maintaining consistency of relationships
- Automatic support for storing objects in external data repositories
- Several store options (SQL, XML, binary)

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Already at Your Fingertips…

Lots of information and samples

Documentation

- Core Data Programming Guide
- NSPersistentDocument Core Data Tutorial
- Low-Level Core Data Tutorial
- Core Data Objective-C Reference (and update)

/Developer/Examples/CoreData

- OutlineEdit
- Stickies
- Tile Puzzle
- Event Manager

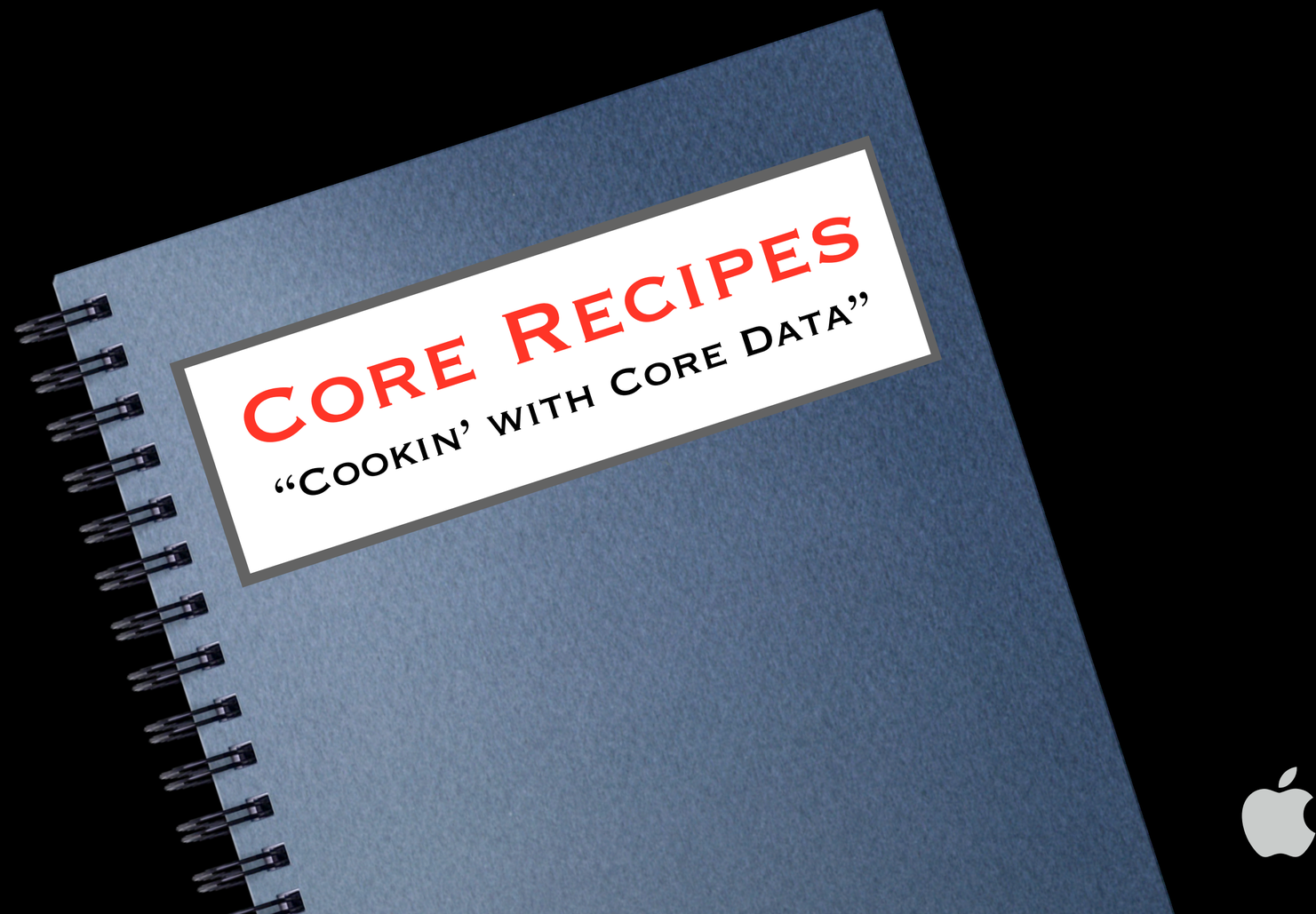
WWDC Sample Code

“CoreRecipes”

- Hands-on samples (for tomorrow’s session)
  - Command-line and GUI applications

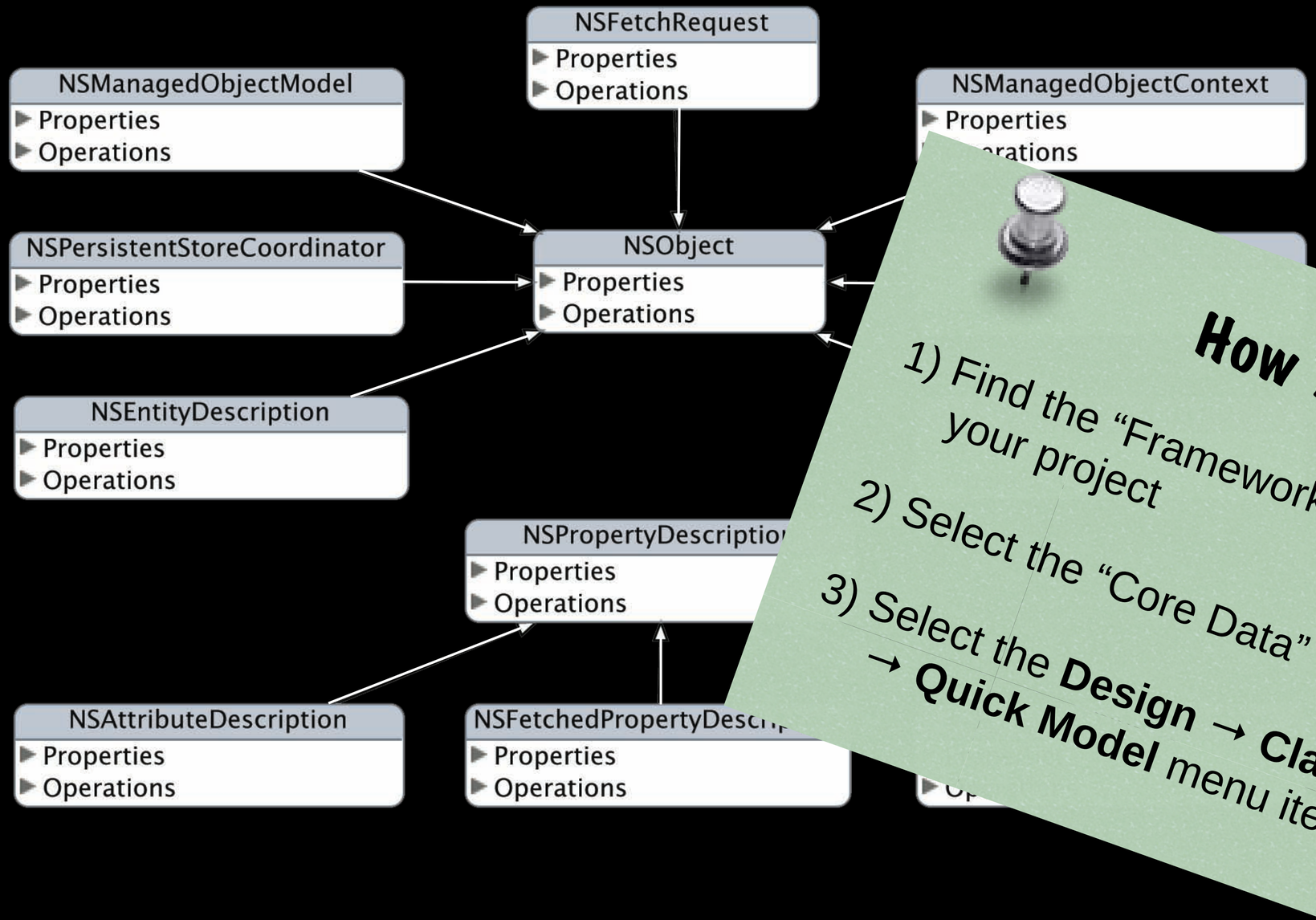
- CoreRecipes application
  - Core Data essentials
  - Cocoa bindings
  - Spotlight integration
  - Automator action

- Data migration sample



Use Xcode's *QuickModel* feature!

Visually investigate the Core Data class hierarchy



Related Sessions

Implementing Core Data	The Presidio Thursday 9:00 AM
Core Data Lab	Applications Lab Thursday 12:00 PM
Getting the Most Out of Cocoa Bindings	SOMA Thursday 2:00 PM
Core Data Lab	Applications Lab Thursday 3:30 PM
Cocoa and Core Data Feedback Forum	North Beach Friday 9:00 AM
Advanced Core Data Usages	The Marina Friday 2:00 PM

Who to Contact

Matthew Formica
Cocoa/Core Data Evangelist
mformica@apple.com

Cocoa Development
Apple Email Discussion List
cocoa-dev@lists.apple.com

Bug Reporter
Apple Bug Tracking System
http://bugreport.apple.com

Q&A

Matthew Firlik
Manager, Core Data

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference